

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-172657

(43)Date of publication of application : 23.06.2000

.....
(51)Int.Cl. G06F 15/177

G06F 13/00

G06F 15/16

.....
(21)Application number : 10-349067 (71)Applicant : FUJITSU LTD

(22)Date of filing : 08.12.1998 (72)Inventor : SEGUCHI YOSHIYUKI
YAMASHIMA HIROYUKI

.....
(54) SYSTEM AND METHOD FOR DISTRIBUTED PROCESSING, COMPUTER-
READABLE RECORDING MEDIUM WITH PROGRAM FOR COMPUTER TO
EXECUTE THE SAME METHOD RECORDED THEREIN, SERVER DEVICE AND
CLIENT DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a distributed processing system capable of comfortably utilizing a service from a server regardless of difference in the classes of clients.

SOLUTION: A service program for executing service is divided into a server control module required for executing the service on the side of server, a display control module for controlling display on a display device provided to a client, an initial setting module prepared for each client class so as to define a method for acquiring the required service

module from the server to the client and a common module with which an execution position does not depend on the client and the server. When the resource lack of the client is detected in the case of executing this service program, the particle activation of the common module is requested to the server, and when the resource lack of the server is detected, the partial activation of the common module is requested to the other server.

LEGAL STATUS [Date of request for examination] 29.10.2001

[Date of sending the examiner's decision of rejection] 10.05.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

*** NOTICES ***

JPO and INPIT are not responsible for any

damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the DPS equipped with the server equipment each other connected through the network, and client equipment said server equipment Two or more modules which can be performed according to the individual for performing offer and utilization of service, The initialization module for acquiring a module required being a kind of said module and for said client equipment using service by this client equipment side, It has a service provision means to offer service by performing said two or more modules. Said client equipment Said required module is acquired from said server equipment by activation of the initialization module which acquired and acquired said initialization module from said server equipment. The distributed processing system characterized by having a service utilization means to use service by performing the acquired module.

[Claim 2] Said server equipment is equipped with the common module which can be performed in the execution environment of the both sides of said server equipment and said client equipment as a kind of said module. Said service provision means In the condition of performing said common module, when said server equipment changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. The distributed processing system according to claim 1 characterized by requiring starting of said common module by the side of these other server equipments from other server equipments which have this function.

[Claim 3] Said server equipment is equipped with the common module which can be performed in the execution environment of the both sides of said server equipment and said client equipment as a kind of said module. Said service utilization means In the condition of performing said common module, when said client equipment changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. The distributed processing system according to claim 1 or 2 characterized by requiring starting of said common module by the side of this server equipment from said server equipment.

[Claim 4] When said module is updated, said server equipment to the client

equipment of the point which distributed this module It has an advice means of updating to notify renewal of said updated module. Said client equipment Server equipment of any one publication of claim 1-3 characterized by having an updating module acquisition means to acquire said updated module from said server equipment when renewal of said updated module is notified.

[Claim 5] In the distributed-processing approach in the distributed processing system equipped with the server equipment each other connected through the network, and client equipment The initialization module distribution step which distributes the initialization module for acquiring the module which said server equipment needs for using service according to the demand of said client equipment by this client equipment side, The distributed-processing approach that said initialization module is acquired, and said client equipment performs the acquired initialization module, and is characterized by including the initialization module execute step which performs the module which acquired and acquired said required module from said server equipment.

[Claim 6] In the distributed-processing approach in the distributed processing system equipped with the server equipment each other connected through the network, and client equipment The server common module execute step to

which said server equipment performs the common module which can be performed in the execution environment of the both sides of said server equipment and other server equipments, When said server equipment changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. The distributed-processing approach characterized by including the server common module activate request step which requires starting of said common module by the side of these other server equipments from server equipment besides the above.

[Claim 7] In the distributed-processing approach in the distributed processing system equipped with the server equipment each other connected through the network, and client equipment The client common module execute step to which said client equipment performs the common module which can be performed in the execution environment of the both sides of said server equipment and said client equipment, When said client equipment changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. The distributed-processing approach characterized by including the client common module activate request step which requires starting of said common module by the side of this server equipment from said server equipment.

[Claim 8] In the distributed-processing approach in the distributed processing system equipped with the server equipment each other connected through the network, and client equipment When the module which said server equipment needs for using service according to the demand of said client equipment is distributed The module distribution place registration step which registers the address of the client equipment of a distribution place into a module distribution place table, The advice step of renewal of a module which notifies renewal of said updated module to the address point of the client equipment registered into said module distribution place table when said module is updated, The distributed-processing approach characterized by said client equipment containing the updating module acquisition step which acquires said updated module from said server equipment for advice of renewal of said updated module to a carrier beam case.

[Claim 9] In the record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed processing system equipped with the server equipment each other connected through the network, and client equipment and in which computer reading is possible The initialization module distribution step which distributes the

initialization module for acquiring a module required using service in said client equipment according to the demand of said server equipment to said client equipment by this client equipment side, Said required module is acquired from said server equipment by performing the initialization module which acquired and acquired said initialization module in said client equipment. The record medium which recorded the program for performing the initialization module execute step which performs the acquired module and in which computer reading is possible.

[Claim 10] In the record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed processing system equipped with the server equipment each other connected through the network, and client equipment and in which computer reading is possible The server common module execute step which performs the common module which can be performed in said server equipment in the execution environment of the both sides of said server equipment and said client equipment, The server common module activate request step which requires starting of said common module by the side of these other server equipments from other server equipments when said server equipment changes into the

resource lack condition by lowering of a throughput, buildup of a load, etc., The record medium which recorded the program for making it perform and in which computer reading is possible.

[Claim 11] In the record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed processing system equipped with the server equipment each other connected through the network, and client equipment and in which computer reading is possible The client common module execute step which performs the common module which can be performed in said client equipment in the execution environment of the both sides of said server equipment and said client equipment, When said client equipment changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. The record medium which recorded the program for performing the client common module activate request step which requires the startup of said common module by the side of this server equipment from said server equipment and in which computer reading is possible.

[Claim 12] In the record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed

processing system equipped with the server equipment each other connected through the network, and client equipment and in which computer reading is possible When a module required to use service in said client equipment is distributed from said server equipment The module distribution place registration step which registers the address of the client equipment of a distribution place into a module distribution place table, The advice step of renewal of a module which notifies renewal of said updated module to the address point of the client equipment registered into said module distribution place table when said module is updated, The record medium which recorded the program for performing the updating module acquisition step which makes said updated module acquire from said server equipment when directions of acquisition of said updated module are notified to said client equipment and in which computer reading is possible.

[Claim 13] In the server equipment which is connected to a network and builds the DPS with client equipment Two or more modules which can be performed according to the individual for performing offer and utilization of service, The initialization module for acquiring a module required being a kind of said module and for said client equipment using service by this client equipment side,

According to the acquisition demand of said initialization module from said client equipment, the initialization module according to the classification of this client equipment is distributed. Server equipment characterized by having a service provision means to offer service by distributing the module demanded by performing this initialization module in said client equipment, and performing said two or more modules.

[Claim 14] In the execution environment of the both sides of said server equipment and said client equipment, it has the common module which can be performed as a kind of said module. Said service provision means In the condition of preparing the resource surveillance which acquires the throughput of said server equipment, and the information on loaded condition for every fixed time amount, and performing said common module When the information acquired by said resource surveillance shows the resource lack condition by lowering of a throughput, buildup of a load, etc. Server equipment according to claim 13 characterized by requiring starting of said common module by the side of these other server equipments from other server equipments which have this function.

[Claim 15] the selection of a common module set as the object of an activate

request in the condition that said service provision means is performing said two or more common modules until the resource lack condition of said server equipment is canceled -- carrying out -- said -- others -- the server equipment according to claim 14 characterized by repeating the demand of starting of said selected common module to server equipment.

[Claim 16] said service provision means -- said -- others -- the case where starting of said common module is required from server equipment -- said common module -- starting -- said -- others -- the server equipment according to claim 14 or 15 characterized by permitting access to said started common module from server equipment or said client equipment.

[Claim 17] It is server equipment of any one publication of claim 14-16 which is equipped with the module activation location table which registers the activation positional information of said server equipment which starts said common module, and is characterized by said service provision means accessing said common module based on said activation positional information registered into said module activation location table.

[Claim 18] The server equipment of any one publication of claim 13-17 characterized by to have the module distribution place table which registers the

address of the client equipment of the point which distributed said module, and an advice means of updating notify renewal of said updated module to the address point of the client equipment registered into said module distribution place table when said module is updated in said server equipment.

[Claim 19] The client equipment acquire the initialization module for acquiring a module required using the service which said server equipment offers in the client equipment which is connected to a network and builds the DPS with server equipment from said server equipment from said server equipment, and carry out having had a service utilization means use service by performing the module which acquired a required module and acquired by activation of the initialization module which acquired as the description.

[Claim 20] Said initialization module is client equipment according to claim 19 characterized by choosing the module acquired from said server equipment among modules required to use the service which performs examination of the throughput of said client equipment, or loaded condition, and said server equipment offers according to these results of an investigation.

[Claim 21] It is client equipment according to claim 19 or 20 which is equipped with the module managed table which manages the module acquired from said

server equipment, and said service utilization means acquires this module from said server equipment when the module required to use service for said module managed table is not registered, and is characterized by to register the acquired module into said module managed table.

[Claim 22] Said service utilization means acquires the common module which can be performed from said server equipment in the execution environment of the both sides of said server equipment and said client equipment as a kind of a module required to use service. In the condition of preparing the resource surveillance which acquires the throughput of said client equipment, and the information on loaded condition for every fixed time amount, and performing said common module When the information acquired by said resource surveillance shows the resource lack condition by lowering of a throughput, buildup of a load, etc. Client equipment of any one publication of claim 19-21 characterized by requiring starting of said common module by the side of this server equipment from said server equipment.

[Claim 23] Said service utilization means is client equipment according to claim 22 characterized by choosing the common module set as the object of an activate request, and repeating the demand of starting of said selected common

module to said server equipment in the condition of performing said two or more common modules until the resource lack condition of said client equipment is canceled.

[Claim 24] It is client equipment according to claim 22 or 23 which is equipped with the module activation location table which registers the activation positional information of said server equipment which starts said common module, and is characterized by said service utilization means accessing said common module based on said activation positional information registered into said module activation location table.

[Claim 25] Client equipment of any one publication of claim 19-24 characterized by having an updating module acquisition means to acquire said updated module from said server equipment for advice of renewal of the module updated from said server equipment to a carrier beam case.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention consists of the servers and clients which were connected through the network. In the distributed processing system and the distributed-processing approach of providing a user with service, the record medium which recorded the program which makes a computer perform the approach and in which computer reading is possible, server equipment, and client equipment The distributed processing system which reduces the server at

the time of performing a service program required in order to realize service especially, the load in the both sides of a client, and the maintenance service of a service program, It is related with the distributed-processing approach, the record medium which recorded the program which makes a computer perform the approach and in which computer reading is possible, server equipment, and client equipment.

[0002]

[Description of the Prior Art] In recent years, utilization of the distributed processing system which makes it possible to receive desired service through a network is general by improvement in maintenance of a communication link infrastructure, and the spread of LAN (Local Area Network) or the Internet. It was established as optimal network connection method when it consists of two or more client equipments (a client is only called hereafter) which a user uses especially, and server equipment (a server is only called hereafter) which offers service, the client-server method which can receive desired service when a client accesses a server is set on a network and various service resources are shared or managed.

[0003] On the other hand, the magnitude of the network which can connect a

user, i.e., a client, has expanded further between the networks with which the Internet was dotted in an area or in the company by interconnecting from LAN to WAN (Wide Area Network). Furthermore, the number of the users using a network, i.e., the number of the clients which belong in each network, is increasing, and buildup of the load of a server and communication link traffic poses a problem with amplification of the magnitude of the above-mentioned network.

[0004] For this reason, in current, two or more servers are prepared according to the service kind offered in a small-scale network, and there is an inclination for distribution of the load in service activation to be performed. For example, on a network, servers, such as a database server which offers database service, a file server which carries out package management of the file, a mail server which manages an electronic mail, and a computation server which makes large-scale and complicated computation a speciality, distribute, and are arranged.

[0005] Thus, while progress of the software about the service transfer on a network is seen, progress to the hardware of the equipment connected to a network, especially the variety which can serve as a client is desired. the communication link interface for mainly connecting a client with CPU (Central

Processing Unit) which performs an operating system and various application programs in a network, the store which memorizes the program performed by CPU, and data, input devices, such as a keyboard and a mouse, and displays, such as CRT and a liquid crystal display, -- since -- it is constituted. Although a server is also the same configuration as a client, the peripheral device for offering various services may be connected further.

[0006] That specification is divided roughly by the class of CPU, and the configuration of a client can be operated as a stand-alone by adopting the operating system suitable for this CPU, when it is many.

[0007] Therefore, the want which connects with a network and uses the activity of a shared resource as a drawing wax between the clients of a different hardware specification arises that it is necessary. That is, even if it is from the client of what kind of hardware specification, a demand of the user who wants to receive the same service from a server increases. In order to meet this demand, the service program for accessing the server which offers service in addition to performing the communication link which used the same communications protocol as the communications protocol adopted in the network, i.e., carrying out a communication link interface in common, is needed. Since this service

program is performed on a client, it is necessary to prepare it separately according to the hardware specification of a client, especially an operating system.

[0008] Generally, the function required for service is beforehand registered into the server and the client as the above-mentioned service program, this service program absorbed the difference in the interface between a hardware specification of a different kind or an operating system of a different kind, and conventional distributed groupware, a conventional distributed network service, etc. have realized service provision from a server to a client.

[0009] The case where a client acquires desired data from the database of a server as an example to which a client receives service in below from a server is explained. In this case, the server used as an access place is a database server which is equipped with stores, such as a large hard disk, and is building the database on the storage region of this store. In addition, a client accesses this database server and carries beforehand the service program (a retrieval program is called hereafter) for searching and acquiring desired data.

[0010] First, a client starts the above-mentioned retrieval program. In addition, in this starting phase, connection with a database server should already be

established. The retrieval program includes display processing, as a result of displaying the query input process which receives the input of retrieval conditions (query), and the retrieval result transmitted from a server generally. A user uses the input device with which the client was equipped, and makes the input of a query complete by advancing selection of the category shown by the input and retrieval program of a keyword about the data to want.

[0011] If query input process is completed, a retrieval program will transmit the information on the inputted query to a database server, and will require retrieval and a transfer of data. The service program corresponding to the retrieval program of a client is performed as a demon, and a database server specifies the data which a user wants promptly from a vast quantity of data stored in the store of a server based on this query information, if the information on the query from a client is received. Continuing, a database server transmits the specified data towards a client with a demand.

[0012] And the retrieval program of a client receives the data transmitted from the database server, and displays this data on a display. Thereby, a user can acquire desired data. Thus, a client is only carrying the comparatively lightweight program of a retrieval program as a service program, and can use the database

which needs a huge capacity.

[0013] Moreover, the client/server system of disclosure is proposed by JP,10-21081,A as other examples of the distributed processing system in the former. This client/server system divides the application program (service program) performed in a client into two or more modules which can be performed according to an individual, and saves the application program which consists of two or more of these modules on the storage of a server. And when a client wants to access this server and to use service, only the module of a part required for activation among the application programs saved at the server is downloaded to the store of a client, and this module is performed.

[0014] In the module under activation, when activation of the module of further others is needed, the module is downloaded from a server. Under the present circumstances, when the total capacity of the module downloaded in the client exceeds a predetermined capacity, the module to have exceeded is deleted from the storage of a client. Thereby, also when a mass application program needs to be performed in a client, pressing resources, such as storage of a client, is lost.

[0015] Therefore, in the conventional distributed processing system explained

above, while being able to use the shared resource on a network, by carrying out a communication link interface in common between the clients of various specifications, connection with a network can be established and the service which a server offers can be used.

[0016]

[Problem(s) to be Solved by the Invention] However, in the conventional DPS, the service program registered into the server and the client was fixed, when [of service] an addition and modification of a function were performed in part, it needed to change the whole service program registered into the server or the client, and it was not able to perform efficient maintenance.

[0017] Moreover, since the service program was generally clearly distinguished as the program for servers, and a program for clients, it was difficult to shift the activation location of service processing and to distribute a load according to the throughput and function of a client or a server.

[0018] Moreover, although the user could choose the service program registered into a client in the conventional DPS according to the resource and function of a client, this restricts service realizable for every client, and had the problem of the service limit depending on a client that no services which a server offers could

be used.

[0019] Furthermore, in the conventional distributed processing system, generally, the user or the system administrator had to install beforehand the service program according to the model of the client for every client, and there was an inclination which enlarges the burden of a user or a system administrator.

[0020] Moreover, also in case the functional addition and modification to a service program were made in relation to the above-mentioned need for install, the whole service program of a server and client equipment needed to be re-installed, and the maintenance to the function according to individual in a service program was difficult. Furthermore, there was a problem that it was difficult for all users to grasp the updating stage of a service program.

[0021] Moreover, in conventional distributed groupware or a conventional distributed network service, although the multi-user system for two or more users with one client equipment was adopted in many cases, even if it was the case where the functions of the service used for every user differed, it needed to hold to the client by having made all functions into the service program, and was not necessarily what used the resource of a client efficiently.

[0022] Although the client/server system of the disclosure to JP,10-21081,A

solved the problem of maintenance over the function according to the problem of the install activity of the service program mentioned above, and individual in a service program, and the problem of the resource of a client since it carried out the modularization of the service program and only the module required for activation downloaded it to the client, it was not what can grasp the updating stage of a service program promptly.

[0023] Moreover, in this client/server system, in case a client receives service, it is necessary to surely perform download of the module from a server. therefore, the time amount for this download needs at least by the condition that CPU of a client can perform a module -- having -- a new module -- one after another -- ** -- in performing a service program which is called, the load of a client not only increases, but a throughput falls and it invites buildup of the load of a server, and communication link traffic.

[0024] Furthermore, in this client/server system, when effective use of a resource only to storage was enabled and a client accessed two or more servers simultaneously, it was not a load to CPU, or a thing in consideration of pressure of the nonvolatile memory used as the location for activation of CPU, and it was not necessarily what raises the performance of the whole network containing

other servers and clients.

[0025] This invention is made in view of the above, and aims at offering the distributed processing system which can use the service from a server comfortably irrespective of the difference in the classification of a client.

[0026]

[Means for Solving the Problem] In order to solve the technical problem mentioned above and to attain the object, it sets to invention of claim 1. The distributed processing system equipped with the server equipment (server) each other connected through the network, and client equipment (client) Two or more modules which a server can perform according to the individual for performing offer and utilization of service (services module), The initialization module for acquiring services modules (DCM etc.) required being a kind of a services module and for a client using service by the client side, A service provision means to offer service by performing two or more services modules (server control module etc.) (with "server demon starting processing" mentioned later) To "initialization module acquisition processing" and the server side processing in "initialization module transaction", with fairly A preparation and a client acquire an initialization module from a server. A required services module is acquired

from a server by activation of the acquired initialization module. A service utilization means to use service by performing the acquired module (with "client demon starting processing" mentioned later) It is characterized by preparing for "initialization module acquisition processing" and the client side processing in "initialization module transaction" fairly.

[0027] According to invention of this claim 1, a server accepts the demand of a client. The initialization module which differ for every classification of a client and can acquire a module required for activation of service from a server is distributed to a client. A client By being able to acquire only a required services module from a server in a client, and performing this acquired services module by acquiring and performing this initialization module from a server Since service can be used, even if it is the case where the client from which a hardware specification and an operating system differ is connected to a network, only by acquiring the initialization module according to a client Since the initialization module to which the service offered by the server can use for same between clients, and corresponds with a service utilization means in a client is automatically acquirable A user can use easily the service which a server offers, without being conscious of the classification of a client.

[0028] In invention of claim 2 moreover, a distributed processing system according to claim 1 A server is equipped with the common module which can be performed in the execution environment of the both sides of a server and a client as a kind of a services module. furthermore, the condition (step S of "service starting processing" mentioned later -- for 115s) that the service provision means is performing this common module It sets fairly at step S143s of step S123s of "module execution environment shift processing in a client", or "module execution environment shift processing in a server." When a server changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. (equivalent to "processing of a system monitoring thread" mentioned later) It is characterized by what (equivalent to step S145s1 of "module execution environment shift processing in a server") starting of the common module by the side of other servers is demanded for from this server and other servers which have this function.

[0029] By having the common module which does not ask a server about an execution environment according to invention of this claim 2 Since an activation location can be shifted so that the common module which should be performed in this server may originally be performed in other servers to the resource lack

condition of a server During activation of a service program, for example, by access of many clients When the resource of a server is pressed, the common module currently performed in the server can be performed now in other servers which have allowances in a resource, and lowering of the throughput by the lack of a resource of a server can be avoided.

[0030] In invention of claim 3 moreover, a distributed processing system according to claim 1 or 2 A server is equipped with the common module which can be performed in the execution environment of the both sides of a server and a client as a kind of a services module. Furthermore, the service utilization means of a client sets this common module in the condition (equivalent to step S112c of "service starting processing") of performing by the client side. When a client changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. (equivalent to "processing of a system monitoring thread") It is characterized by what (equivalent to step S124c of "module execution environment shift processing in a client") starting of the common module by the side of the server is demanded for from a server.

[0031] According to invention of this claim 3, by a server being equipped with the common module which can be performed in the execution environment of the

both sides of a server and a client Since an activation location can be shifted so that the common module which should be performed in a client may originally be performed in a server to the resource lack condition of a client During activation of a service program, for example, by activation of other application programs etc. When the resource of a client is pressed, the common module currently performed in the client can be performed now in a server, and lowering of the throughput by the lack of a resource of a client can be avoided.

[0032] In invention of claim 4 moreover, the distributed processing system of any one publication of claim 1-3 When renewal of a services module arises, a server to the client of the point which distributed the services module It has an advice means of updating (equivalent to server side processing of "a services module update process" mentioned later) to notify renewal of the updated services module. A client When renewal of the services module updated from the server is notified, it is characterized by having an updating module acquisition means (equivalent to client side processing of a "services module update process") to acquire the updated services module from a server.

[0033] According to invention of this claim 4, since that advice is further performed to the client of the point which distributed that services module in the

past, it becomes unnecessary for a user to be conscious of the updating stage of the module especially in a server in a client by the ability of a server notifying renewal of a services module to a client, and an update process of an efficient services module is attained.

[0034] Moreover, the distributed-processing approach adopted in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 5 The module which needs a server to use service according to the demand of a client (it is a services module) The initialization module distribution step which distributes the initialization module for acquiring the DCM etc. by the client side (equivalent to the server side processing in "initialization module acquisition processing"), A client acquires this initialization module and performs the acquired initialization module. From a server, a required services module is acquired and it is characterized by including the initialization module execute step (equivalent to "initialization module acquisition processing" and the client side processing in "initialization module transaction") which performs the acquired services module.

[0035] Because according to invention of this claim 5 a client acquires the initialization module which differ for every classification of a client and acquires a

services module required for activation of service from a server and performs this In a client, only a required services module is acquirable from a server, and by activation of this acquired services module, since utilization of service is enabled Even if it is the case where the client from which a hardware specification and an operating system differ is connected to a network, a user can use the service offered by the server like the case where other clients are used.

[0036] Moreover, the distributed-processing approach adopted in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 6 the server common module execute step (step S of "service starting processing" -- for 115s) to which a server performs the common module which can be performed in the execution environment of the both sides of a server and other servers At step S143s of step S123s of "module execution environment shift processing in a client", or "module execution environment shift processing in a server" with fairly When a server changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. (equivalent to "processing of a system monitoring thread") It is characterized by including the server common module activate request step

(equivalent to step S145s1 of "module execution environment shift processing in a server") which requires starting of the common module by the side of other servers from other servers.

[0037] When the resource lack condition of a server arises in a server during activation of the common module which does not ask an execution environment according to invention of this claim 6, it is required that this common module should be performed in other servers. Since the activation location of a common module can be shifted, for example During activation of a service program, by access of many clients When the resource of a server is pressed, the common module currently performed in the server can be performed now in other servers which have allowances in a resource, and lowering of the throughput by the lack of a resource of a server can be avoided.

[0038] Moreover, the distributed-processing approach adopted in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 7 The client common module execute step to which a client performs the common module which can be performed in the execution environment of the both sides of a server and a client (equivalent to step S112c of "service starting processing"), When a client

changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. (equivalent to "processing of a system monitoring thread") It is characterized by including the client common module activate request step (equivalent to step S124c of "module execution environment shift processing in a client") which requires starting of the common module by the side of the server from a server.

[0039] When the resource lack condition of a client arises in a client during activation of the common module which does not ask an execution environment according to invention of this claim 7, it is required that this common module should be performed in a server. Since the activation location of a common module can be shifted, for example During activation of a service program, by activation of other application programs etc. When the resource of a client is pressed, the common module currently performed in the client can be performed now in a server, and lowering of the throughput by the lack of a resource of a client can be avoided.

[0040] Moreover, the distributed-processing approach adopted in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 8 the services module (an

initialization module --) which needs a server to use service according to the demand of a client When the DCM etc. is distributed The module distribution place registration step which registers the address of the client of a distribution place into a module distribution place table (server data control table) (equivalent to step S102s of the step S93s and "initialization module transaction" of "initialization module acquisition processing"), When a services module is updated, at the address point of the client registered into the module distribution place table The advice step of renewal of a module which notifies renewal of the updated services module (equivalent to server side processing of a "services module update process"), A client is characterized by including the updating module acquisition step (equivalent to client side processing of a "services module update process") which acquires the module updated from the server in advice of renewal of the updated module at the carrier beam case.

[0041] Since that advice is further performed to the client of the point which is referring to a module distribution place table, and distributed that services module in the past, it becomes unnecessary for a user to be conscious of the updating stage of the module especially in a server in a client by the ability of a server notifying renewal of a services module to a client according to invention of

this claim 8, and an update process of an efficient services module is attained.

[0042] The record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 9 and in which computer reading is possible moreover, from a server A module required to use service in the client according to the demand of a client (it is a services module) The initialization module distribution step which distributes the initialization module for acquiring the DCM etc. by the client side (equivalent to the server side processing in "initialization module acquisition processing"), A required services module is acquired from a server by performing the initialization module which acquired and acquired the initialization module in the client. It is characterized by performing the initialization module execute step (equivalent to "initialization module acquisition processing" and the client side processing in "initialization module transaction") which performs the acquired module.

[0043] According to invention of this claim 9, by installing this record medium in a server and a client Because a client acquires the initialization module which differ for every classification of a client and acquires a services module required

for activation of service from a server and performs this In a client, only a required services module is acquirable from a server, and by activation of this acquired services module, since the program which enables utilization of service can be performed Even if it is the case where the client from which a hardware specification and an operating system differ is connected to a network, a user can use the service offered by the server like the case where other clients are used.

[0044] Moreover, the record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 10 and in which computer reading is possible the server common module execute step (step S of "service starting processing" -- for 115s) which performs the common module which can be performed in a server in the execution environment of the both sides of a server and a client At step S143s of step S123s of "module execution environment shift processing in a client", or "module execution environment shift processing in a server" with fairly When a server changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. (equivalent to "processing of a

system monitoring thread") It is characterized by performing the server common module activate request step (equivalent to step S145s1 of "module execution environment shift processing in a server") which requires starting of the common module by the side of other servers from other servers.

[0045] According to invention of this claim 10, by installing this record medium in a server and a client When the resource lack condition of a server arises, it is required that this common module should be performed in other servers in a server during activation of the common module which does not ask an execution environment. Since the program which can be shifted in the activation location of a common module can be performed During activation of a service program, for example, by access of many clients When the resource of a server is pressed, the common module currently performed in the server can be performed now in other servers which have allowances in a resource, and lowering of the throughput by the lack of a resource of a server can be avoided.

[0046] Moreover, the record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 11 and in which computer reading is

possible The client common module execute step which performs the common module which can be performed in a client in the execution environment of the both sides of a server and a client (equivalent to step S112c of "service starting processing"), When a client changes into the resource lack condition by lowering of a throughput, buildup of a load, etc. (equivalent to "processing of a system monitoring thread") It is characterized by performing the client common module activate request step (equivalent to step S124c of "module execution environment shift processing in a client") which requires starting of the common module by the side of the server from a server.

[0047] According to invention of this claim 11, by installing this record medium in a server and a client When the resource lack condition of a client arises, it is required that this common module should be performed in a server in a client during activation of the common module which does not ask an execution environment. Since the program which can be shifted in the activation location of a common module can be performed During activation of a service program, for example, by activation of other application programs etc. When the resource of a client is pressed, the common module currently performed in the client can be performed now in a server, and lowering of the throughput by the lack of a

resource of a client can be avoided.

[0048] The record medium which recorded the program which makes a computer perform the distributed-processing approach in the distributed processing system equipped with the server mutually connected through the network, and the client in invention of claim 12 and in which computer reading is possible moreover, from a server a services module (an initialization module --) required to use service in a client When the DCM etc. is distributed The module distribution place registration step which registers the address of the client of a distribution place into a module distribution place table (server data control table) (equivalent to step S102s of the step S93s and "initialization module transaction" of "initialization module acquisition processing"), When a services module is updated, at the address point of the client registered into the module distribution place table The advice step of renewal of a module which notifies renewal of the updated services module (equivalent to server side processing of a "services module update process"), When directions of acquisition of the updated services module are notified to a client It is characterized by performing the updating module acquisition step (equivalent to client side processing of a "services module update process") which makes the services module updated from the

server acquire.

[0049] According to invention of this claim 12, by installing this record medium in a server and a client A server can notify renewal of a services module to a client.

Further the advice Since the program which could be carried out by referring to a module distribution place table to the client of the point which distributed the services module in the past can be performed In a client, it becomes unnecessary for a user to be conscious of the updating stage of the module especially in a server, and the update process of an efficient services module of him is attained.

[0050] Moreover, the server equipment (server) which is connected to a network and builds the DPS with client equipment (client) in claim 13 Two or more modules which can be performed according to the individual for performing offer and utilization of service (services module), The initialization module for acquiring services modules (DCM etc.) required being a kind of a services module and for a client using service by the client side, According to the acquisition demand of the initialization module from a client, the initialization module according to the classification of the client is distributed. The services module demanded by performing the initialization module in a client is

distributed. It is characterized by having a service provision means (equivalent to "server demon starting processing", and "initialization module acquisition processing" and the server side processing in "initialization module transaction") to offer service by performing a services module.

[0051] According to invention of this claim 13, it can respond only by changing the content of the initialization module, also when modification of a services module required for the service program for using service by the client side since a server distributes to a client the initialization module which enables distribution of a different services module for every classification of a client etc. arises, and management of many services modules which exist for every classification of a client becomes easy.

[0052] In claim 14 moreover, server equipment (server) according to claim 13 In the execution environment of the both sides of a server and a client, it has the common module which can be performed as a kind of a services module. In the condition that a service provision means prepares the resource surveillance (equivalent to "processing of a system monitoring thread") which acquires the throughput of a server, and the information on loaded condition for every fixed time amount, and is performing the common module When the information

acquired by resource surveillance shows the resource lack condition by lowering of a throughput, buildup of a load, etc. It is characterized by what (equivalent to step S145s1 of "module execution environment shift processing in a server") starting of the common module by the side of other servers is demanded for from this server and other servers which have this function.

[0053] By having the common module which does not ask a server about an execution environment according to invention of this claim 14 Since an activation location can be shifted so that the common module which should be performed in this server may originally be performed in other servers to the resource lack condition of a server When the lack of a resource of a server arises, the common module currently performed in the server can be performed now in other servers which have allowances in a resource, and lowering of the throughput of a server can be avoided.

[0054] Moreover, in claim 15, a server according to claim 14 is characterized by what the demand of starting of the common module which the service provision means chose the common module set as the object of an activate request, and chose to other servers in the condition of performing two or more common modules until the resource lack condition of a server was canceled is repeated

for (equivalent to "module execution environment shift processing in a server").

[0055] Since only the common module by which the lack of a resource is canceled among two or more common modules currently performed is set as the object of an activate request to other servers according to invention of this claim 15, the big pressure to the resource of the server of the activate request point is avoidable.

[0056] Moreover, in claim 16, a server according to claim 14 or 15 is characterized by what (equivalent to "module execution environment shift processing in a client", and "module execution environment shift processing in a server") access to the common module which started the common module and was started from other servers or clients is permitted for, when starting of a common module is required of a service provision means from other servers.

[0057] Directly since [according to invention of this claim 16 / to the client which needs activation of not only the server that emitted the activate request for access to the common module by which the activate request was carried out but its common module] it is possible, utilization of a common module by which the activate request was carried out is realized easily.

[0058] Moreover, in claim 17, the server of any one publication of claim 14-16 is

equipped with the module activation location table (server data-control table) which registers the activation positional information of the server which starts a common module, and is characterized by what (equivalent to "services module access processing" mentioned later) a service provision means accesses a common module for based on the activation positional information registered into the module activation location table.

[0059] According to invention of this claim 17, since the activation location of a common module is managed on a module activation location table, access of a desired common module becomes easy by referring to this module activation location table.

[0060] In claim 18 moreover, the server of any one publication of claim 13-17 The module distribution place table which registers the address of the client of the point which distributed the services module (server data control table), An advice means of updating to notify renewal of the services module updated by the address point of the client registered into the module distribution place table when a services module was updated in a server (equivalent to server side processing of a "services module update process"), It is characterized by preparation *****.

[0061] Since advice of updating is performed only to a client [need / the services module updated by referring to a module distribution place table / to be acquired] according to invention of this claim 18, an update process of an efficient services module is attained.

[0062] Moreover, the client equipment (client) which is connected to a network and builds a distributed processing system with server equipment (server) in claim 19 The initialization module for acquiring services modules (DCM etc.) required using the service which a server offers from a server is acquired from a server. A service utilization means to acquire a required services module and to use service by performing the acquired services module by activation of the acquired initialization module (with "client demon starting processing") It is characterized by preparing for "initialization module acquisition processing" and the client side processing in "initialization module transaction" fairly.

[0063] According to invention of this claim 19, a client Only by acquiring and performing an initialization module from a server with a service utilization means By processing of this initialization module, since only a required services module is acquirable from a server in a client Even if it is the case where the client from which a hardware specification and an operating system differ is connected to a

network, while the service offered by the server can use same between clients A user can use easily the service which a server offers, without being conscious of the classification of a client.

[0064] Moreover, in claim 20, an initialization module performs examination of the throughput of a client, or loaded condition (equivalent to step S104c of "initialization module transaction"), and a client according to claim 19 is characterized by what (equivalent to step S105c of "initialization module transaction") the services module acquired from a server among services modules required to use the service which a server offers according to the results of an investigation is chosen for.

[0065] According to invention of this claim 20, the examination device in which the function and throughput of a client are investigated for the service utilization means performed in a client is established. Since the services module which acquires this examination device from a server according to the results of an investigation performed and acquired at the time of activation of an initialization module can be chosen For example, when results of an investigation show lowering of the throughput of a client Only the services module which can be performed is acquired and performed and the service utilization which becomes

possible to request activation of the services module of an and also [it is the need] from a server etc., and does not have stress in service activation in a client is attained.

[0066] In claim 21 moreover, a client according to claim 19 or 20 It has the module managed table (client data control table) which manages the services module acquired from the server. When the services module which a service utilization means needs for using service for a module managed table is not registered The services module is acquired from a server and it is characterized by what (equivalent to step S107c of "initialization module transaction") the acquired services module is registered into a module managed table for.

[0067] Since that services module is acquired from a server and it registers with a client data control table when the services module required to use service for a client data control table is not registered according to this claim 21, a client can equip attached storage etc. only with a minimum services module required for utilization of service, and can realize the effective activity of the resource of a client.

[0068] In claim 22 moreover, the client of any one publication of claim 19-21 In the execution environment of the both sides of a server and a client, the common

module which can be performed is acquired from a server as a kind of the services module which a service utilization means needs for using service. The resource surveillance (equivalent to "system monitoring thread processing") which acquires the throughput of a client and the information on loaded condition for every fixed time amount is prepared. In the condition (equivalent to step S112c of "service starting processing") of performing the common module When the information acquired by resource surveillance shows the resource lack condition by lowering of a throughput, buildup of a load, etc. It is characterized by what (equivalent to step S124c of "module execution environment shift processing in a client") starting of the common module by the side of the server is demanded for from a server.

[0069] In the condition of performing the common module which a client does not ask about an execution environment according to this claim 22 Since an activation location can be shifted so that the common module which should be performed in this client may originally be performed in a server to the resource lack condition of a client When the lack of a resource of a client arises, the common module currently performed in the client can be performed now in the server which has allowances in a resource, and lowering of the throughput of a

client can be avoided.

[0070] moreover, in claim 23, in the condition that the service utilization means be perform two or more common modules, a client according to claim 22 choose the common module set as the object of an activate request, and be characterize by what the demand of starting of said selected common module be repeat for to said server equipment (equivalent to "module execution environment shift processing in a client") until the resource lack condition of a client be cancel.

[0071] Since only the common module by which the lack of a resource is canceled among two or more common modules currently performed is set as the object of an activate request to a server according to this claim 23, the big pressure to the resource of the server of the activate request point is avoidable.

[0072] Moreover, in claim 24, a client according to claim 22 or 23 is equipped with the module activation location table (client data control table) which registers the activation positional information of the server which starts a common module, and is characterized by what (equivalent to "services module access processing") a service utilization means accesses a common module for based on the activation positional information registered into the module

activation location table.

[0073] According to this claim 24, since the activation location of a common module is managed on a module activation location table, access of a desired common module becomes easy by referring to this module activation location table.

[0074] Moreover, in claim 25, the client of any one publication of claim 19-24 is characterized by having an updating module acquisition means (equivalent to client side processing of a "services module update process") to acquire the services module updated from the server at the carrier beam case in advice of renewal of the services module updated from the server.

[0075] Since according to this claim 25 the updated services module is acquired with an updating module acquisition means when renewal of a services module is notified from a server, in a client, it becomes unnecessary for the user himself to be conscious of the updating stage of the module in a server, and the update process of an efficient services module of him is attained.

[0076]

[Embodiment of the Invention] Below, the gestalt of operation of the DPS concerning this invention, the distributed-processing approach, server

equipment, and client equipment is explained at a detail based on a drawing. In addition, this invention is not limited by the gestalt of this operation. Here, the gestalt of the concrete operation which performs theoretic explanation about each processing which operates first in the DPS concerning this invention, the distributed-processing approach, server equipment, and client equipment, and operates by each [these] processing continuously is explained.

[0077] The DPS concerning this invention consists of one or more servers and one or more clients which are connected through communication line networks, i.e., a network, such as a LAN cable, the telephone line, or a dedicated line, and is a system by which a client can use the service which a server mainly offers as a shared resource on a network. By connecting two or more servers to a network especially, the load in service activation is distributed and the environment which the service which a client receives can perform without stress is offered.

[0078] Moreover, this distributed processing system is carrying out a communication link interface in common, and two or more sorts of clients from which a hardware specification and an operating system differ may be connected.

[0079] The service in here is that a server provides a client with various data to

the demand of a client, and are database service, computation service, etc. Therefore, it is necessary to have a service program for the servers for performing the target service according to the demand from a client, and, as for a server, a client also needs to be equipped with the service program for the clients for requiring service from a server.

[0080] the communication link interface for connecting a client with CPU which performs an operating system, the above-mentioned service program for clients, and other application programs in a network, the store which memorizes the program performed by CPU, and data, input devices, such as a keyboard and a mouse, and displays, such as CRT and a liquid crystal display, -- since -- it is constituted and the usual computer system can be used. Moreover, although a server is the same configuration as a client, the peripheral device for offering having further two or more CPUs and enabling high-speed processing corresponding to a multiprocessor, and a mass store and various services may be connected. In addition, the above-mentioned store contains memory, such as RAM and ROM, a hard disk drive unit, and the drive equipment whose loading of a record medium with still more nearly dismountable IC card memory, floppy disk, magneto-optic disk, CD-ROM, etc. is enabled.

[0081] In the DPS concerning this invention, the above-mentioned service program can be constituted from two or more modules which can be performed according to an individual, and both the service program for servers and the service program for clients can be performed per module. Moreover, the module group which constitutes the service program for clients can be distributed to a client from a server as service data, and maintenance and management of these modules (a services module is called hereafter) are mainly performed by the server.

[0082] Drawing 1 is the explanatory view showing the configuration of module of the service data which are offered, and which were prepared for every service. A server control module required in order to perform service activation to service data by the server side in drawing 1 , The DCM which performs the display control to the display with which the client was equipped, and performs input/output control of a graphical user interface (GUI) and others, The initialization module which has the function for it to be prepared for every client classification, and to define the acquisition approach from the server of a services module required for a client, and to perform examination of the throughput and loaded condition of a client, It is divided into the common module

which furthermore does not depend for the activation location of a services module on a client and a server.

[0083] In the service data shown in drawing 1 , the service program for servers and the service program for clients which were mentioned above are constituted by combining the services module to need, and can be equipped with two or more each services modules (a server control module, the DCM, an initialization module, common module).

[0084] For example, the service program for servers consists of some server control modules, and the service program for clients consists of one initialization module, some DCM, and some common modules.

[0085] Drawing 2 is the explanatory view showing the system activation gestalt of a server. The server demon SD 1 to whom a server performs registration of the connection request from two or more clients, delivery of a services module, and service control of a server to the bottom of the execution environment of an operating system SOS resides permanently. In this condition, if the service activation demand from a client occurs, the thread for performing the services module of the service program SSP 1 (Service-1) which corresponds in the server demon SD 1 will be generated, and this services module will be

performed.

[0086] Moreover, the system monitoring thread STH 1 which manages the throughput and loaded condition of a server serially is contained in the server demon SD 1, and system monitoring to a server is given to him with a fixed time interval. Moreover, the server equips the store with the server data control table SDT 1 which manages the information on the services module registered into a server and a client, and data with the thread in the server demon SD 1 are outputted if needed and inputted with reference to this server data control table SDT 1.

[0087] In addition, the server demon SD 1 performed I/O of data and an instruction to the operating system SOS, and the operating system SOS has established connection with I/O of data with a network, i.e., a client, and other servers through the socket API for a communication link (Application Program Interface).

[0088] the service name of service which a server provides with the server data control table SDT 1, the services module name of a services module required to perform this service, and service module data including the information about the number of versions and running state of each services module -- since -- it is

constituted. Drawing 3 is the explanatory view having shown the service module data to each services module of the server data control table SDT 1.

[0089] In drawing 3 , the service module data in the server data control table SDT 1 For example, corresponding "services module name", the "default service name" which shows a corresponding service name, The "address-for-delivery client address list" in which the client to which the server delivered the services module is shown, the classification (a server control module and the DCM --) of a services module The "module classification" which shows an initialization module and a common module, The "plat-form classification" which shows the operating system or hardware specification of a client, The "version number" of services modules, the "status flag" which shows the activation location of a services module, The "access ID at the time of local activation" which shows the access code to a services module in case a "status flag" shows local activation "The server address at the time of remote activation" which shows the address of a server with which a services module in case a "status flag" shows "remote activation" belongs, It consists of data, such as the "access ID" etc. which shows the access code to a services module in case a "status flag" shows "remote activation."

[0090] Drawing 4 is the explanatory view showing the system activation gestalt of a client. The client demon CD 1 to whom a client performs the communication link with a server to the bottom of the execution environment of an operating system COS resides permanently. Here, an operating system COS does not necessarily need to be the same as the operating system SOS of a server.

[0091] In this condition, if the service activation demand from a user occurs, the thread for performing the services module of the service program CSP 1 (Service-1) which corresponds in the client demon CD 1 will be generated, and this services module will be performed.

[0092] Moreover, the system monitoring thread CTH1 which manages the throughput and loaded condition of a client serially is contained in the client demon CD 1, and system monitoring to a client is given to him with a fixed time interval. Moreover, the client is equipped with the client data control table CDT1 which manages the information on the services module which acquires from a server to a store and is registered into it by the client, and data with the thread in the client demon CD 1 are outputted if needed and inputted with reference to this client data control table CDT1.

[0093] In addition, the client demon CD 1 performed I/O of data and an

instruction to the operating system COS, and the operating system COS has established I/O of data with a network, i.e., the connection with a server, through the socket API for a communication link like a server.

[0094] the service name of service by which the client data control table CDT1 was registered into the client, the services module name of a services module required to perform this service, and service module data including the information about the number of versions and running state of each services module -- since -- it is constituted. Drawing 5 is the explanatory view having shown the service module data to each services module of the client data control table CDT1.

[0095] In drawing 5 , the service module data in the client data control table CDT1 consists of data, such as an "acquisition server address" etc. which shows the address of the server which is the acquisition place of the "services module name", a "default service name", a "version number", a "status flag", the "access ID at the time of local activation", "the server address at the time of remote activation", "Access ID", and the services module which were explained to drawing 3 .

[0096] Below, in the activation gestalt of the server and client which were shown

in drawing 2 and drawing 4 , flow until service is performed is explained according to a flow chart.

[0097] (Starting processing of a client demon) As for the terminal unit used as a client, in many cases, the client demon CD 1 is first performed by starting and coincidence under the environment of an operating system COS. Thereby, the thread by which the client demon CD 1 was always generated as a client demon is performed, and I/O of the data according to the content of activation of each thread is performed.

[0098] Drawing 6 is the flow chart which showed the processing performed in starting of the client demon CD 1. In drawing 6 , the client demon CD 1 is first started by the task bootstrap by which automatic activation is carried out at starting of an operating system COS and coincidence (step S61). The client demon CD 1 receives the message from a server, or transmits the message demand from the services module performed by the client to a server.

[0099] The DCM which controls the event about the service selection and service activation of a carbon button depression, text input, etc. which are performed because a user uses an input device, the initialization module which enables acquisition of the services module according to client classification, and

the common module independent of the activation location of a client or a server are contained in the services module performed by the client.

[0100] The system monitoring thread CTH1 which performs the monitor of the throughput of a client or loaded condition is generated and performed with starting of this client demon CD 1 (step S62). Thus, after the system monitoring thread CTH1 resides permanently, the list of services which can offer a server of services registered into the current server, i.e., a list, is acquired (step S63).

[0101] The thread for performing the service list acquisition from a server in the client demon CD 1 is generated actually, and this thread requires acquisition of a service list from an operating system COS. An operating system COS (in detail the communication link socket API) requires distribution of a service list from a server with an accessible client.

[0102] Thus, the thread which acquires a service list also functions as a services module, and serves as an indispensable module in a client. Therefore, a user can acquire a service list from a server not only the time of starting of the client demon CD 1 but if needed.

[0103] If a service list is distributed from a server, the GUI display according to a service list is performed on the indicating equipment of a client, access through

the input device by the user or server access will await, and the above-mentioned service list acquisition thread or the thread of the DCM registered beforehand will change in the condition (step S64).

[0104] Here, server access shows the instruction to the services module currently performed on message sending which a server performs to a client, or a client, or transmission of data.

[0105] (Starting processing of a server demon) Starting processing of the server demon SD 1 is explained below mentioning later the processing after step S64 shown in drawing 6 . As for the terminal unit used as a server, the thread by which the server demon SD 1 is in a resident condition, and was generated as a server demon is usually always performed under the environment of an operating system SOS. Since a server always needs to receive access from a client, every time it removes the cases of being special, such as a maintenance service, it may think that it is always activation status. For this reason, although starting processing of the server demon SD 1 is not performed rarely, either, here explains the case where starting of the server demon SD 1 by actuation of the time of starting of a server or a system administrator is performed.

[0106] Drawing 7 is the flow chart which showed the processing performed at the

time of starting of the server demon SD 1. In drawing 7 , the server demon SD 1 is started first (step S71). The server demon SD 1 receives the message from a client or other servers, or transmits the message demand from the services module performed within a server to a client or other servers.

[0107] The server control module started by service and the service activate request from a client of a resident mold and the common module independent of the activation location of a client or a server are contained in the services module performed by the server.

[0108] The system monitoring thread STH 1 which performs management of the throughput of a server or loaded condition is generated and performed with starting of this server demon SD 1 (step S72). And after the system monitoring thread STH 1 resides permanently, it changes from access from a client, or other servers to the waiting receptacle condition for access (step S73).

[0109] (Processing of a system monitoring thread) Below, the executive operation (step S72) of the system monitoring thread STH 1 in activation (step S62) and "starting processing of a server demon" of the system monitoring thread CTH1 in "starting processing of the client demon CD 1" mentioned above is explained. In addition, both the system monitoring threads CTH1 and STH1

show processing of the same flow.

[0110] Drawing 8 is the flow chart which showed processing by the system monitoring thread. A system monitoring thread is arranged by a client and the server, respectively, and is generated by the client demon CD 1 and the server demon SD 1. Continuing, a system monitoring thread performs the check of the throughput of a system (a server or client), or loaded condition in which this system monitoring thread resides permanently (step S81). Here, as a check of a throughput, examination of the data accumulation capacity of memory or a store, the number of openings of a communication link socket required for the communication link with a server or a client, the permissible dose of thread generation, etc. is raised, and measurement by implementation of a CPU operating ratio or a benchmark test is raised as a check of loaded condition.

[0111] It continues and it is judged as a result of the check of the throughput and loaded condition of step S81 whether a system shows lowering and the remarkable loaded condition of a throughput, i.e., the condition in short of a resource, (step S82). In step S82, when the condition of the lack of a resource of a system is shown, it is judged further whether the activation location of this system monitoring thread is a client (step S84).

[0112] In step S84, module execution environment shift processing for clients (step S120) which will be later mentioned if the activation location of a system monitoring thread is a client is performed, and if the activation location of a system monitoring thread is a server, module execution environment shift processing for servers (step S140) mentioned later will be performed.

[0113] On the other hand, when the lack of a resource is not detected in step S82, fixed time amount awaits, it will be in a condition (step S83), and processing of step S81 will be again performed after this fixed time amount progress. That is, the processing of a system monitoring thread shown in drawing 8 shows that module execution environment shift processing according to the location (a server or client) where this system monitoring thread is performed is performed, when the throughput and loaded condition of a system are checked for every fixed time amount and the result of that check shows the lack of a resource.

[0114] (Initialization module acquisition processing) Next, service selection by the user is performed to the user access / waiting receptacle condition for server access of step S64 shown in drawing 6 , and the initialization module acquisition processing performed as a result is explained. Drawing 9 is the flow chart which

showed initialization module acquisition processing.

[0115] In drawing 9 , a user chooses desired service from the service list by which it was indicated by GUI first using input devices, such as a mouse and a keyboard, on the indicating equipment of a client (step S91c). And an initialization module required to perform selected service is searched from the above-mentioned client data control table CDT1 (step S92c).

[0116] The judgment of whether the initialization module which continues and corresponds is registered into the client data control table CDT1 is performed (step S93c). In step S93c, when the corresponding initialization module is registered into the client data control table CDT1, in order to check the number of versions of the initialization module registered into the server, the information on the initialization module corresponding to the initialization module described above among the initialization modules registered into the server is acquired (step S94c).

[0117] The information acquisition demand of the initialization module accompanied by the services module name which expresses the corresponding initialization module as a procedure of processing of step S94c is sent to a server. And a client receives the information on the initialization module which is

distributed from a server to this initialization module information acquisition demand and which answers namely, corresponds.

[0118] The initialization module information which the client received is constituted like the service module data shown in drawing 3 , performs the comparison of **, the number information of versions included in this data, and the initialization module information registered into the client, for example, number information of versions included in the service module data shown in drawing 5 , and judges whether the number of versions is in agreement (step S95c).

[0119] In step S95c, since it is necessary to change the initialization module which corresponds in a client when the number of versions is not in agreement (i.e., when a value with the number of versions of the initialization module registered into the client older than the number of versions of the initialization module registered into the server is shown), the acquisition demand of an initialization module is sent to a server.

[0120] Moreover, in step S93c, also when the corresponding initialization module is not registered into the client data control table CDT1, the acquisition demand of an initialization module is sent to a server. In addition, the numerical

information which specifies the service name corresponding to the initialization module made into the object and the plat-form classification (client classification) which specifies the hardware specification of a client or the class of operating system is included in the initialization module acquisition demand.

[0121] A server specifies a service name and client classification by the argument (the above-mentioned numerical information) which the initialization module acquisition demand which received from the client shows (step S 91s). It continues and a server chooses from the server data control table STD 1 the initialization module corresponding to the service name and client classification which were specified in step S91s (step S 92s).

[0122] And a server distributes the initialization module chosen in step S92s to a client with an initialization module acquisition demand. A server registers the address of the client which had the initialization module acquisition demand in the "address-for-delivery client address list" among the service module data corresponding to the distributed initialization module in the server data control table SDT 1 after distribution of an initialization module (step S 93s).

[0123] On the other hand, the client (in detail the client demon CD 1) which received the initialization module registers the received initialization module into

the client data control table CDT1. Under the present circumstances, it registers with the "acquisition server address" in the service module data which shows the address of the server of an acquisition place to drawing 5 (step S96c).

[0124] Continuing, a client generates the thread of the initialization module registered in step S96c (step S97c), and starts this initialization module (step S98c).

[0125] In addition, initialization module acquisition processing in which it explained above may be included in the service program constituted by the service list acquisition thread mentioned above or this thread, and may be made to perform by registering beforehand the services module for performing initialization module acquisition processing into a client.

[0126] (Initialization module transaction) The processing which follows next step S98c shown in drawing 9 , i.e., processing of an initialization module, is explained. Drawing 10 is the flow chart which showed processing by the initialization module. In drawing 10 , an initialization module extracts a services module required for activation of service from the information defined as the interior first (step S101c).

[0127] And an initialization module searches the services module extracted in

step S101c from the client data control table CDT1 (step S102c). In step S102c, when the services module which serves as an object for retrieval from the client data control table CDT1 can be specified (i.e., when the services module used as the object for retrieval is registered into the client data control table CDT1), the number of versions of the services module is checked (step S103c).

[0128] The check of the number of versions of the services module in step S103c needs to send an acquisition demand of the information on the services module which corresponds among the services modules registered into the server to the server, in order to acquire first the number of versions of the services module registered into the server.

[0129] The number information of versions is acquired from the services module information from which reception and an initialization module received the services module information to which a client is distributed from a server after dispatch of this services module information acquisition demand, and which answers namely, corresponds. In addition, it is also possible to set to perform this services module information acquisition demand, and to acquire the information on two or more corresponding services modules by transmission and reception once, when the information on two or more services modules is

required.

[0130] On the other hand, to the services module registered into the client, the number information of versions is acquirable from the service module data corresponding to the target services module in the client data control table CDT1.

In addition, in step S102c, when the services module used as the object for retrieval is not registered into the client data control table CDT1, processing of step S103c is skipped and processing of the following step S104c is performed.

[0131] In step S104c, the check of the function of a current client, a throughput, and loaded condition is performed by the system call to an operating system COS. Although the system monitoring thread CTH1 mentioned above can also perform the system check to this client and an acquisition demand of the information on whether an initialization module has a system in a resource lack condition to the system monitoring thread CTH1 can be performed in that case, in order to perform the quick status check of a system especially, the system check original with an initialization module which used the system call shall be performed here.

[0132] It judges whether it continues and an initialization module needs to acquire the services module extracted in step S101c from a server (step S105c).

In step S105c, when a services module needs to be acquired from a server and the services module is not specifically registered into the client data control table CDT1 in step S102c, or when the number information of versions on both services module acquired in step S103c differs, the acquisition demand of a services module is sent to a server.

[0133] Using the information included in the services module acquisition demand which received from the client, a server specifies a services module and chooses the specified services module from the server data control table SDT 1 (step S 101s).

[0134] And a server distributes the services module chosen in step S101s to a client with a services module acquisition demand. A server registers the address of the client which had the services module acquisition demand in the "address-for-delivery client address list" among the service module data corresponding to the distributed services module in the server data control table SDT 1 after distribution of a services module (step S 102s).

[0135] On the other hand, the client (in detail the client demon CD 1) which received the services module judges first whether the services module which received is registered into the client data control table CDT1 (step S106c). In the

system check in step S104c, or the system monitoring thread CTH1, processing of step S106c shows the processing for requesting from a server activation of the services module which should be performed, especially a common module, when a resource lack condition is shown.

[0136] In step S106c, when registering a services module (i.e., when the lack of a resource of a client is not detected), the services module which received from the server is registered into the client data control table CDT1. Under the present circumstances, it registers with the "acquisition server address" in the service module data which shows the address of the server of an acquisition place to drawing 5 (step S107c).

[0137] In step S106c, when not registering a services module (i.e., when the lack of a resource of a client is detected), module execution environment shift processing for clients (step S120 of drawing 8) mentioned later is performed.

[0138] And after module execution environment shift processing for clients is performed, starting processing of service in which it explains below after termination of step S107c is performed.

[0139] (Service starting processing) Starting processing of the service chosen in step S91c of drawing 9 mentioned above next is explained. Drawing 11 is the

flow chart which showed service starting processing. In drawing 11 , the client demon CD 1 performs first the activate request of the server control module which performs server control to a server by the service activate request from a user. By this demand, a server starts the corresponding server control module (step S 111s).

[0140] At this time, lowering of a client throughput is remarkable, and when not registering a services module in the case of an overload condition, i.e., step S106c of "initialization module transaction" mentioned above, further, or when the lack of a resource is newly detected by the system monitoring thread CTH1, the client demon CD 1 performs a common module activate request, in order to request from a server starting of the common module which should be performed essentially in the end of a local.

[0141] Therefore, the judgment (step S 112s) of the existence of the common module activate request following processing of step S111s is needed. Step S In 112s, when a common module activate request exists, the check of the throughput and loaded condition in this server is performed first (step S 113s). That is, the server demon SD 1 acquires the information which shows the resource condition of a server from the system monitoring thread STH 1.

[0142] It continues and the judgment with a resource insufficient [a system] is performed by the information acquired by step S113s (step S 114s). Step S When the lack of a resource is shown in 114s, module execution environment shift processing for servers for requesting activation of a common module with an activate request from the server of further others (step S140) is performed.

[0143] Step S When the lack of a resource is not detected in 114s, the server demon SD 1 starts the common module of which starting was required (step S 115s), and sets up an effective address required for access to the common module, Access ID, and a status flag (refer to "the server address at the time of remote activation" of the service module data shown in drawing 3 , "Access ID", and a "status flag") (step S 116s).

[0144] And advice of the completion of service starting including the information on the effective address which showed completion of starting with a common module when there is starting of service, i.e., the activate request of a server control module and a common module, to the client with an activate request, and was set up in step S116s, Access ID, and a status flag is distributed. Then, the server demon SD 1 changes in the waiting receptacle condition for client access (step S 117s). This shows the condition that starting of the service program for

servers (a server control module, common module) was completed.

[0145] The client demon CD 1 will start the DCM corresponding to the service to perform, if advice of the completion of service starting is received from a server (step S111c). Only the common module which can be performed is started according to the system resource information on the client acquired by system monitoring thread CTH1 grade among the common modules which should be continued and performed (step S112c).

[0146] and the service module data of a common module required for service activation looks the client demon CD 1 like [the client data control table CDT1], respectively, he sets on it, and registers "the server address at the time of local activation", "the server address at the time of remote activation", "Access ID", a "status flag", etc. into it (step S113c). Thereby, it is shown that starting of the service program for clients (the DCM, common module) was completed.

[0147] Therefore, a user can perform access (access to the detailed service program for servers) to a server with the service program for these clients, and can receive the service which a server offers.

[0148] (Module execution environment shift processing in a client) Below, when the lowering and the overload condition of the throughput of a system by the

system monitoring thread CTH1 are detected in a client, the processing which shifts the execution environment of a common module to a server is explained. That is, this processing corresponds to the module execution environment shift processing for clients of step S120 in the processing of a system monitoring thread shown in drawing 8 .

[0149] Drawing 12 is the flow chart which showed the module execution environment shift processing in a client. In addition, this module execution environment shift processing is offered by the module execution environment shift program beforehand registered in the client, and by being generated in the client demon CD 1, that thread shall reside in a system permanently and shall be performed.

[0150] In drawing 12 , the client demon CD 1 extracts a one-eyed common module from the client data control table CDT1 among the common modules under present activation first (step S121c). And the client demon CD 1 judges whether the extract of a common module was possible in step S121c (step S122c).

[0151] When the extract of a common module is not possible, namely, when the common module required for activation is not registered into the client data

control table CDT1 any more in step S122c, this module execution environment shift processing is ended. Moreover, in step S122c, it judges whether when the extract of a common module is possible, the information on a status flag is acquired from the service module data of the extracted common module, and this status flag shows "local activation" (step S123c).

[0152] In order to search the following common module in the client data control table CDT1 since it is not necessary to shift an execution environment when the common module a status flag does not indicate "local activation" to be in step S123c and which was case [the common module] namely, extracted is performed in the server, it returns to processing of step S121c again.

[0153] When the common module a status flag indicates "local activation" to be in step S123c and which was case [the common module] namely, extracted is performed in the client, the common module activate request for requesting activation of this common module to a server is performed (step S124c).

[0154] A server will check the throughput and loaded condition in this server first, if the common module activate request from a client is received (step S 121s). That is, the server demon SD 1 acquires the information which shows the resource condition of a server from the system monitoring thread STH 1.

[0155] It continues and the judgment with a resource insufficient [a system] is performed by the information acquired in step S121s (step S 122s). Step S When the lack of a resource is detected in 122s, module execution environment shift processing for servers for requesting activation of a common module with an activate request from the server of further others (step S140) is performed.

[0156] Step S When the lack of a resource is not detected in 122s, the server demon SD 1 starts the common module by which the activate request was carried out (step S 123s), and sets up an effective address required for access to the common module, Access ID, and a status flag (refer to "the server address at the time of remote activation" of the service module data shown in drawing 3 , "Access ID", and a "status flag") (step S 124s).

[0157] And advice of the completion of common module starting including the information on the effective address which showed completion of starting of a common module and was set up in step S124s, Access ID, and a status flag is distributed to a client with an activate request.

[0158] If advice of the completion of common module starting is received from a server, the client demon CD 1 suspends activation of the common module under local activation, and when are recording fields are insufficient, he will delete

common modular programming. Continuing, the client demon CD 1 changes the status flag of the service module data of the target common module into "remote activation" in the client data control table CDT1 (step S125c).

[0159] And the client demon CD 1 performs registration of "the server address at the time of remote activation", "Access ID", and a "status flag" on the client data control table CDT1 in each of the service module data of a common module required for service activation (step S126c).

[0160] Furthermore, the client demon CD 1 judges whether a system is the lack of a resource (lowering or the overload condition of a throughput) by acquiring system resource information from the system monitoring thread CTH1 (step S127c). When the lack of a resource is not detected in step S127c, this module execution environment shift processing is ended.

[0161] When the lack of a resource is detected in step S127c, processing of step S121c is repeated again. That is, this module execution environment shift processing is repeated until the lack of a resource of a system is canceled. Moreover, since it is owned jointly between the DPS concerning this invention by other services modules and the client demon CD 1, even if a common module is the case where the activation location of a common module is changed, with it, it

needs to guarantee access to a common module similarly.

[0162] Drawing 13 is the flow chart which showed services module access processing. The client demon CD 1 or other services modules perform services module access processing shown in drawing 13 in advance of access to a services module required for activation. In drawing 13 , the client demon CD 1 or other services modules search the service module data of the corresponding services module (here especially common module) from the client data control table CDT1 first (step S131c). In addition, the setup of the services module by the initialization module mentioned above by the time it performed this services module access processing shall be completed.

[0163] It continues, the client demon CD 1 or other services modules acquire the information on a status flag from the service module data searched in step S131c, and it judges whether a status flag shows "local activation" (step S132c). When a status flag is not "local activation" in step S132c (i.e., when a status flag shows "remote activation"), the processing to this services module is required from the server which "the server address at the time of remote activation" contained in service module data shows. In addition, the information on the "access ID" for accessing the corresponding services module is also included in

advice of this demand.

[0164] The server demon SD 1 receives the services module processing demand from a client, specifies the "access ID" which serves as a processing object from this demand, is using this the "access ID", and attains access of the corresponding services module (step S 131s).

[0165] And when a status flag shows "local activation" in step S132c, "the access ID at the time of local activation" is acquired from the service module data of the corresponding services module, and access of the corresponding services module is attained by using this access ID (step S133c).

[0166] Therefore, according to this services module access processing, even if the activation location of a common module is a client and it is a server, neither other services modules nor the client demon CD 1 is influenced. Moreover, since the activation location of a common module is prescribed by "the server address at the time of remote activation", and "Access ID", the server which performs remote activation is not limited to the server from which the client acquired the common module.

[0167] (Module execution environment shift processing in a server) Below, when lowering and the overload condition of the throughput of a system are detected

by the system monitoring thread STH 1 in a server, the processing which shifts the execution environment of a common module to other servers is explained. That is, this processing corresponds to the module execution environment shift processing for servers of step S140 in the processing of a system monitoring thread shown in drawing 8 .

[0168] Drawing 14 is the flow chart which showed the module execution environment shift processing in a server. In addition, this module execution environment shift processing is offered by the module execution environment shift program beforehand registered in the server, and by being generated in the server demon SD 1, that thread shall reside in a system permanently and shall be performed.

[0169] In drawing 14 , the server demon SD 1 judges first whether the activate request of a services module has occurred to this server (step S141s1). Step S When the activate request of a services module has not occurred in 1 for 141s, a one-eyed services module is extracted from the server data control table SDT 1 among the services modules under current activation (step S142s1). And the server demon SD 1 judges whether the extract of a services module was possible in step S142s1 (step S143s1).

[0170] Step S When the extract of a services module is not possible, namely, when the services module required for activation is not registered into the server data control table SDT 1 any more in 1 for 143s, this module execution environment shift processing is ended. Moreover, in step S143s1, it judges whether when the extract of a services module is possible, the information on a status flag is acquired from the service module data of the extracted services module, and this status flag shows "local activation" (step S144s1).

[0171] Step S In order to search the following services module in the server data control table SDT 1 since it is not necessary to shift an execution environment when the services module a status flag does not indicate "local activation" to be and which was case [the services module] namely, extracted is performed in other servers in 1 for 144s, it returns to processing of step S142s1 again.

[0172] Step S When the services module a status flag indicates "local activation" to be in 1 for 144s and which was case [the services module] namely, extracted is performed in this server (the 1st server), the services module activate request for requesting activation of this services module to other servers (the 2nd server) is performed (step S145s1).

[0173] Other servers (the 2nd server) will check the throughput and loaded

condition in this server first, if a services module activate request is received from the above-mentioned server (step S141s2). That is, the server demon of the 2nd server acquires the information which shows the resource condition of the 2nd server from the system monitoring thread of the 2nd server.

[0174] It continues and the judgment with a resource insufficient [a system] is performed by the information acquired in step S141s2 (step S142s2). Step S When the lack of a resource is detected in 2 for 142s, the module execution environment shift processing for servers for requesting activation of a services module with an activate request from the server (the 3rd server) of further others, i.e., this module execution environment shift processing, is recursively performed to the 3rd server.

[0175] Step S When the lack of a resource is not detected in 2 for 142s, the server demon of the 2nd server starts the services module by which the activate request was carried out (step S143s2), and sets up an effective address required for access to the services module, Access ID, and a status flag (refer to "the server address at the time of remote activation" of the service module data shown in drawing 3 , "Access ID", and a "status flag") (step S144s2).

[0176] And advice of the completion of services module starting including the

information on the effective address which showed completion of starting of a services module and was set up in step S144s2, Access ID, and a status flag is distributed to the 1st server which emitted the activate request.

[0177] If the server demon SD 1 of the 1st server receives advice of the completion of services module starting from the 2nd server or other servers, activation of the services module under local activation will be suspended. Continuing, the server demon SD 1 of the 1st server changes the status flag of the service module data of the target services module into "remote activation" in the server data control table SDT 1 (step S146s1).

[0178] And the server demon SD 1 of the 1st server performs registration of "the server address at the time of remote activation", "Access ID", and a "status flag" on the server data control table SDT 1 in each of the service module data of a services module required for service activation (step S147s1).

[0179] Furthermore, the server demon SD 1 of the 1st server judges whether a system is the lack of a resource (lowering or the overload condition of a throughput) by acquiring system resource information from the system monitoring thread STH 1 (step S148s1). Step S When the lack of a resource is not detected in 1 for 148s, this module execution environment shift processing is

ended.

[0180] Step S When the lack of a resource is detected in 1 for 148s, processing of step S142s1 is repeated again. That is, this module execution environment shift processing is repeated until the lack of a resource of a system is canceled. Even if it is the case where the activation location of a services module is changed by module execution environment shift processing in the server explained above, access to a services module can be guaranteed by services module access processing shown in drawing 13 .

[0181] (Services module update process) When the services module registered into the server is updated next, the processing whose server demon SD 1 demands updating registration of a services module from a client automatically is explained.

[0182] Drawing 15 is the flow chart which showed the services module update process. In drawing 15 , renewal of the services module registered into the server by the system administrator etc. is performed first (step S 151s). And an address-for-delivery client address list is acquired from the service module data of a services module which became an object for updating in the server data control table SDT 1 by the directions or the server demon SD 1 by a system

administrator etc. (step S 152s).

[0183] The server demon SD 1 distributes the renewal demand of a services module used as advice of renewal of a services module to the client shown in the address-for-delivery client address list acquired in step S152s continuously (step S 153s). The number information of versions on the initialization module according to the client used as the object for distribution is included in this renewal demand of a services module.

[0184] Here, the server demon SD 1 confirms whether the client used as the object for distribution logs on to a network (step S 154s), when the log on condition of a client cannot be checked by the network out of order or terminal un-starting, skips distribution of the renewal demand of a services module to the client as error termination, and repeats the log on check to other clients used as the object for distribution. In addition, by this error termination, to the client to which the renewal demand of a services module was not distributed, the server demon SD 1 performs a log on check for every fixed time amount of a certain, and tries distribution of the renewal demand of a services module.

[0185] It judges whether on the other hand, the client demon CD 1 who received the renewal demand of a services module should change an initialization module

from the service module data of an initialization module in the client managed data table CDT1 by acquiring the number information of versions and performing the comparison with the number information of versions on the initialization module contained in the renewal demand of a services module (step S151c).

[0186] In step S151c, when an initialization module needs to be changed (i.e., when the number of versions of the initialization module contained in the renewal demand of a services module is newer than the number of versions of the initialization module registered into the client managed data table CDT1), the acquisition demand of an initialization module is sent to a server. In addition, the numerical information which specifies the service name corresponding to the initialization module made into the object and the plat-form classification (client classification) which specifies the hardware specification of a client or the class of operating system like "initialization module acquisition processing" mentioned above is included in the initialization module acquisition demand.

[0187] By the numerical information which the initialization module acquisition demand which received from the client shows, a server specifies a service name and client classification and chooses from the server data control table STD 1 the initialization module corresponding to the service name and client

classification which were specified. Continuing, a server distributes this selected initialization module to a client with an initialization module acquisition demand. Moreover, a server registers the address of the client which had the initialization module acquisition demand in the "address-for-delivery client address list" among the service module data corresponding to the distributed initialization module in the server data control table SDT 1 after distribution of an initialization module.

[0188] The client demon CD 1 registers this initialization module into the client data control table CDT1 after acquiring an initialization module from a server. And the client demon CD 1 performs the corresponding initialization module, after registration of the initialization module to the above-mentioned client data control table CDT1 when it does not need modification of an initialization module in step S151c (step S152c).

[0189] Continuing, the client demon CD 1 judges whether it is necessary to acquire the services module defined as the initialization module from a server (step S153c). In step S153c, when the services module which specifically corresponds when a services module needs to be acquired from a server is not registered into the client data control table CDT1, or when the number

information of versions on the registered services module is old, the acquisition demand of a services module is sent to a server.

[0190] Using the information included in the services module acquisition demand which received from the client, a server specifies a services module and chooses the specified services module from the server data control table SDT 1.

And a server distributes the selected services module to a client with a services module acquisition demand. Moreover, a server registers the address of the client which had the services module acquisition demand in the "address-for-delivery client address list" among the service module data corresponding to the distributed services module in the server data control table SDT 1 after distribution of a services module.

[0191] On the other hand, the client which received the services module registers into the client data control table CDT1 the services module which received from the server (step S154c). Under the present circumstances, it registers with the "acquisition server address" in the service module data which shows the address of the server of an acquisition place to drawing 5 . Therefore, as for a user, this services module update process enables it to always enjoy service by the number of versions of the newest services module, without being

conscious of renewal of a services module.

[0192] In addition, the services module update process explained above is performed by registering beforehand the services module for performing for example, a services module update process into a client.

[0193] In order to deepen an understanding more about the distributed processing system of this invention which operates next based on the processing explained above, the gestalt of still more concrete operation is explained.

[0194] (Gestalt 1 of operation) Drawing 16 is the explanatory view having shown the system configuration at the time of the DPS concerning the gestalt 1 of operation service un-starting. In drawing 16, the DPS concerning the gestalt 1 of operation consists of that Server#1 which performs service provision as a server, and Client#1 which receive service as a client carry out a communication link interface in common, and they are connected through a network.

[0195] In addition, actually, although many servers and clients are connected and constituted by the network, the DPS concerning this invention is explained focusing on processing between one a server and one client here, in order to make an understanding easy.

[0196] Here, the logical address on the network of Server#1 is set to #0001, and the logical address on the network of Client#1 is set to #0011. Server#1 received the connection request from a client and it is equipped with the server data control table SDT 1 which performs management of service data with the server demon SD 1 who performs execution control of service. Moreover, Server#1 is in the condition after server demon starting processing shown in drawing 7 was performed, and server access of others [// client access] awaits it, and it is in a condition. Therefore, in the server demon SD 1, the system monitoring thread STH 1 which performs management of the throughput of a client or loaded condition is generated.

[0197] Client#1 performed the communication link with Server#1 and it is equipped with the client data control table CDT1 which manages the service data acquired from the server with the client demon CD 1 who provides a user with an interface. Moreover, Client#1 is in the condition after client demon starting processing shown in drawing 6 was performed, and user access / server access awaits it, and it is in a condition. Therefore, in the client demon CD 1, the system monitoring thread CTH1 which performs management of the throughput of a client or loaded condition is generated.

[0198] Moreover, both Server#1 and Client#1 operate under the execution environment of an operating system OS, and they perform data I/O with a network through the communication link socket API. Here, as mentioned above, since Server#1 and Client#1 are carrying out the communication link interface in common, an operating system OS does not necessarily need to be the same [one].

[0199] Here, the case where Service#1 is performed as one of the services offered by the server is considered. Drawing 17 is the explanatory view showing service data required in order to perform Service#1. That is, the service program for servers and the service program for clients are constituted by the combination of the services module contained in this service data.

[0200] In drawing 17 service data Service#1 It consists of two or more services modules. Specifically Server control-module ServiceModule#A1-An which is performed by the server and performs service control, DCM ServiceModule#B1-Bn which is performed by the client and performs a GUI display and the communication link with a server, Initialization module ServiceModule#C1-Cn by which it performed by the client and examination of the throughput of the acquisition approach of a module required for service

activation or a system and loaded condition was defined, It consists of common module ServiceModule#D1-Dn(s) which do not depend for an execution environment on a server or a client. Moreover, as for each services module, only a required number is prepared depending on every client classification and service.

[0201] First, in the service sheep activation status of Server#1 and Client#1 as shown in drawing 16 , a user chooses desired service from the service list shown on the indicating equipment of Client#1 by client demon starting processing shown in drawing 6 . By this service selection, initialization module acquisition processing shown in drawing 9 is performed, and the acquisition demand of an initialization module to Server#1 is performed. In addition, the initialization module required for activation, DCM, and common module of service applicable to the client data control table CDT1 of Client#1 shall not be registered here.

[0202] And Server#1 returns initialization module ServiceModule#C1 according to the client classification of Client#1 to Client#1. On the other hand, Client#1 performs reception and starting of initialization module ServiceModule#C1 for Server#1 to initialization module ServiceModule#C1. By starting of this initialization module ServiceModule#C1, initialization module transaction shown

in drawing 10 is performed, and acquisition of DCM ServiceModule#B1 required for activation of Service#1 and common module ServiceModule#D1 is performed from the content of a definition in initialization module ServiceModule#C1.

[0203] Here, in Client#1, although Service#1 is performed, common module ServiceModule#D2 was still more nearly required, but since the lack of a resource of Client#1 was detected by the system monitoring thread CTH1, acquisition and starting of common module ServiceModule#D2 shall not be performed, but shall request starting of common module ServiceModule#D2 from Server#1.

[0204] Client#1 performs the activate request of Service#1 to Server#1, and service starting processing shown in drawing 11 is performed. Under the present circumstances, Server#1 starts common module ServiceModule#D2 which had the activate request from Client#1, and it changes in the waiting receptacle condition for client access.

[0205] It continues, starting of DCM ServiceModule#B1 and common module ServiceModule#D1 is performed in Client#1, and starting of Service#1 is completed.

[0206] Drawing 18 is the explanatory view having shown the system configuration after service starting of the DPS concerning the gestalt 1 of operation. In addition to the condition which shows Server#1 in drawing 16 , as a service program for servers which offers Service#1, server control-module ServiceModule#A1 and common module ServiceModule#D2 are prepared, and a thread is generated in the server demon SD 1 in activation of these services modules in drawing 18 .

[0207] Moreover, in addition to the condition which shows Client#1 in drawing 16 , as a service program for clients which offers Service#1, DCM ServiceModule#B1, initialization module ServiceModule#C1, and common module ServiceModule#D1 are prepared, and a thread is generated in the client demon CD 1 in activation of these services modules.

[0208] Below, in the initialization module acquisition processing shown in drawing 9 , the service module data of Server#1 when Server#1 distributes an initialization module to a client is explained. Here, the case where Server#1 distributes initialization module ServiceModule#C1 to two or more clients (Client#1, Client#2, Client#3) especially is considered.

[0209] Drawing 19 is the explanatory view showing the service module data of

initialization module ServiceModule#C1 registered into the server data control table SDT 1 of Server#1 in this case. In drawing 19 , first, "ServiceModule#C1" is registered as a "services module name", and "Service#1" is registered as a "default service name." And the logical address on the network of two or more above-mentioned client Client#1, Client#2, and Client#3 (#0011, #0012, #0013) is registered into the "address-for-delivery client list."

[0210] Moreover, "3" which is the code which shows an initialization module as a "module classification" is registered, and the code "1" which specifies the hardware specification or operating system of Client#1 is registered as a "plat-form classification."

[0211] Furthermore, when "1" is registered as a "version number" and initialization module ServiceModule# C1 is updated, this "version number" is updated and the services module update process shown in drawing 15 is performed. Here, the services module update process to initialization module ServiceModule# C1 is performed as automatic processing which a user does not mind to Client#1 registered into the "address-for-delivery client list", Client#2, and Client#3.

[0212] Below, the service module data of common module ServiceModule#D1 is

explained as an example of the service module data registered into the client data control table CDT1 of Client#1.

[0213] Drawing 20 is the explanatory view showing the service module data of common module ServiceModule#D1 registered into the client data control table CDT1 of Client#1 in this case. In drawing 20 , first, "ServiceModule#D1" is registered as a "services module name", and "Service#1" is registered as a "default service name." And "1" is registered as a "version number" and "#0001" which shows the logical address of the server which distributed this common module ServiceModule#D1 as an "acquisition server address" is registered.

[0214] Furthermore, "1" which is the code which shows local activation, i.e., activation of common module ServiceModule#D1 of Client#1, as a "status flag" is registered, and the access ID required in order to access this common module ServiceModule#D1 as "access ID at the time of local activation" "111" is registered.

[0215] Next lowering or the overload condition of a throughput is detected by the system monitoring thread processing shown in drawing 8 , and the service module data of Server#1 in case services module access processing shown in drawing 13 is performed, and Client#1 is explained. From the condition shown in

drawing 18 , an overload condition is detected in Client#1 and the case where Client#1 performs the activation request of common module ServiceModule#D1 to Server#1 is considered especially here.

[0216] Drawing 21 is the explanatory view having shown the system configuration after service starting of the DPS concerning the gestalt 1 of operation, and shows the condition of having started common module ServiceModule#D1 Server#1 was requested [ServiceModule#D] from Client#1, in the condition which showed especially in drawing 18 . Therefore, as shown in drawing 21 , Client#1 opens common module ServiceModule#D1 from the service program for clients, and Server#1 starts common module ServiceModule#D1.

[0217] If common module ServiceModule#D1 is started in Server#1, the service module data of common module ServiceModule#D1 of the server data control table SDT 1 will be changed. Drawing 22 is the explanatory view showing the service module data of common module ServiceModule#D1 registered into the server data control table SDT 1 of Server#1 in this case.

[0218] In drawing 22 , first, "ServiceModule#D1" is registered as a "services module name", and "Service#1" is registered as a "default service name." And

"#0011" which is the above-mentioned logical address of client Client#1 is registered into the "address-for-delivery client list."

[0219] Moreover, "4" which is the code which shows a common module as a "module classification" is registered, and the code "0" which does not specify a server, a client, a hardware specification, or an operating system is registered as a "plat-form classification."

[0220] Furthermore, "1" is registered as a "version number" and "1" which is the code which shows local activation, i.e., activation of common module ServiceModule#D1 of Server#1, as a "status flag" is registered. And ID "222" required in order to access this common module ServiceModule#D1 as "access ID at the time of local activation" is registered.

[0221] On the other hand, if the module activation in Server#1 is completed also in Client#1, the service module data of common module ServiceModule#D1 in the client data control table CDT1 will be changed. Drawing 23 is the explanatory view showing the service module data of common module ServiceModule#D1 registered into the client data control table CDT1 of Client#1 in this case.

[0222] In drawing 23 , first, "ServiceModule#D1" is registered as a "services module name", and "Service#1" is registered as a "default service name." And

"1" is registered as a "version number" and "#0001" which shows the logical address of the server which distributed this common module ServiceModule#D1 as an "acquisition server address" is registered.

[0223] Furthermore, "2" which is the code which shows remote activation, i.e., activation of common module ServiceModule#D1 by the server, as a "status flag" is registered. And "#0001" which shows the address of Server#1 which is the server which starts this common module ServiceModule#D1 as "a server address at the time of remote activation" is registered, and ID "222" required to access common module ServiceModule#D1 on this Server#1 as "access ID" is registered.

[0224] As explained above, according to the distributed processing system concerning the gestalt 1 of operation By activation of "starting processing of a client demon", "starting processing of a server demon", "initialization module acquisition processing", the "initialization module transaction", and "service starting processing" which were mentioned above Even if it is the case where the client from which a hardware specification and an operating system differ is connected to a network, only by acquiring the initialization module according to a client Since the initialization module to which the service offered by the server

can use for same between clients, and corresponds with the service utilization means of a client is automatically acquirable. It becomes possible to use easily the service which a server offers, without requiring the information of the configuration of module required for service of a user.

[0225] Moreover, by activation of "processing of a system monitoring thread" and "module execution environment shift processing in a client" which were mentioned above, when lowering of the throughput of a client is shown, while being able to start on a server a part of common module currently performed in the client and being able to avoid promptly lowering of the throughput by the lack of a resource of a client, an effective load distribution can be realized.

[0226] Furthermore, by activation of a "services module update process" mentioned above, it becomes unnecessary for especially the user using a client to be conscious of the updating stage of the module in a server, and the update process of a quick module of him is attained in a client.

[0227] Next, the condition corresponding to the content of registration of the service module data of drawing 19 and drawing 20 which were explained in the gestalt 1 of operation, Client#1 [i.e.,], receives Server#1. (Gestalt 2 of operation) In the condition (condition shown in drawing 18) of having performed

the activate request of common module ServiceModule#D2 The case where the activate request of common module ServiceModule#D2 is performed by the lack of a resource of Server#1 to Server#2 which are the server of further others is explained.

[0228] Drawing 24 is the explanatory view having shown the system configuration at the time of service starting of the DPS concerning the gestalt 2 of operation. In drawing 24 , the DPS concerning the gestalt 2 of operation consists of that Server#1 which can offer Service#1 as a server and Server#2, and Client#1 which receive Service#1 as a client carry out a communication link interface in common, and they are connected through a network.

[0229] In addition, actually, although many servers and clients are connected and constituted by the network, the DPS concerning this invention is explained focusing on processing between two a server and one client here, in order to make an understanding easy.

[0230] Here, the logical address on the network of Server#1 and Server#2 is set to #0001 and #0002, respectively, and the logical address on the network of Client#1 is set to #0011. Both Server#1 and Server#2 are in the condition of enabling activation of the service starting processing shown in drawing 11 , and

especially Server#1 is in the condition that module execution environment shift processing shown in drawing 14 was performed by detection of the lack of a resource by the system monitoring thread processing shown in drawing 8 .

[0231] Here, other servers in the module execution environment shift processing shown in drawing 14 correspond to Server#2, and starting of common module ServiceModule#D2 which had the activate request from Client#1 is performed by these Server#2. Therefore, as shown in drawing 24 , in Server#2, as a service program for servers which offers Service#1, common module ServiceModule#D2 is prepared and it is in the condition that the thread was generated in the server demon SD 2 in activation of this services module.

[0232] On the other hand, like the condition which shows in drawing 18 explained in the gestalt 1 of operation, as a service program for clients which offers Service#1, DCM ServiceModule#B1, initialization module ServiceModule#C1, and common module ServiceModule#D1 are prepared, and Client#1 is in the condition that the thread was generated in the client demon CD 1 in activation of these services modules.

[0233] Moreover, Server#1, Server#2, and Client#1 operate under the execution environment of an operating system OS, and they perform data I/O with a

network through the communication link socket API. Here, as mentioned above, since Server#1, Server#2, and Client#1 are carrying out the communication link interface in common, an operating system OS does not necessarily need to be the same [one].

[0234] Below, the service module data of Server#1 and Client#1 in the condition which showed in drawing 24 is explained. The case where Client#1 performed the activation request of common module ServiceModule#D2 to Server#1, and Server#1 performs the activation request of common module ServiceModule#D2 to Server#2 further is considered especially here.

[0235] Drawing 25 is the explanatory view showing the service module data of common module ServiceModule#D2 registered into the server data control table SDT 1 of Server#1 in this case.

[0236] In drawing 25 , first, "ServiceModule#D2" is registered as a "services module name", and "Service#1" is registered as a "default service name." And "4" which is the code which shows a common module as a "module classification" is registered, and the code "0" which does not specify a server, a client, a hardware specification, or an operating system is registered as a "plat-form classification."

[0237] Furthermore, "1" is registered as a "version number" and "2" which is the code which shows remote activation, i.e., activation of common module ServiceModule#D2 by servers other than Server#1, as a "status flag" is registered. And "#0002" which shows the address of Server#2 which are the server which starts this common module ServiceModule#D2 as "a server address at the time of remote activation" is registered, and ID "333" required to access common module ServiceModule#D2 on these Server#2 as "access ID" is registered.

[0238] Drawing 26 is the explanatory view showing the service module data of common module ServiceModule#D1 registered into the client data control table CDT1 of Client#1 in this case.

[0239] In drawing 26 , first, "ServiceModule#D1" is registered as a "services module name", and "Service#1" is registered as a "default service name." And "1" is registered as a "version number" and "#0001" which shows the logical address of the server which distributed this common module ServiceModule#D2 as an "acquisition server address" is registered.

[0240] Furthermore, "2" which is the code which shows remote activation, i.e., activation of common module ServiceModule#D2 by the server, as a "status

flag" is registered. And "#0002" which shows the address of Server#2 which are the server which starts this common module ServiceModule#D2 as "a server address at the time of remote activation" is registered, and ID "333" required to access common module ServiceModule#D2 on these Server#2 as "access ID" is registered.

[0241] As explained above, according to the distributed processing system concerning the gestalt 2 of operation By activation of "module execution environment shift processing in a server" mentioned above When the resource of a server is pressed, a part of common module currently performed in the server While being able to start on other servers which have allowances in a resource and being able to avoid promptly lowering of the throughput by the lack of a resource of a server, an effective load distribution is realizable.

[0242] moreover, the attribute (a services module name --) of the services module which is distributed to a client from a server according to the gestalten 1 and 2 of the operation explained above The server data control table SDT 1 equipped with the service module data by which registration setting out is carried out in a default service name, the number of versions, etc. and a running state (status flag), and the client data control table CDT1 Access to a common module

can be ensured by being able to grasp the activation location of a services module required for activation of service by preparing for a server and a client, respectively, and performing "services module access processing" mentioned above.

[0243] The program (detailed module group which can be performed according to an individual) which performs processing shown in drawing 6 -15 in the gestalten 1 and 2 of the principle explanation and implementation which were explained above In addition, IC card memory, The program which stored in record media, such as a floppy disk, a magneto-optic disk, and CD-ROM, and was recorded on this record medium by installing in server equipment and client equipment The distributed-processing approach mentioned above can be used and, thereby, a distributed processing system can also be built. In addition, this install activity may be performed by downloading between servers as opposed to a client using a communication line from a server.

[0244]

[Effect of the Invention] According to the distributed-processing approach concerning the DPS concerning claim 1 of this invention, and claim 5, as explained as mentioned above, server equipment According to the demand of

client equipment, it differs for every classification of client equipment. The initialization module which can acquire a module required for activation of service from server equipment is distributed to client equipment. And client equipment By acquiring and performing this initialization module from server equipment Since service can be used by being able to acquire only a required services module from server equipment in client equipment, and performing this acquired services module Even if it is the case where the client equipment with which a hardware specification differs from an operating system is connected to a network, only by acquiring the initialization module according to client equipment Since the initialization module to which the service offered by server equipment can use same between client equipment, and corresponds with a service utilization means in client equipment is automatically acquirable A user can use easily the service which server equipment offers, without being conscious of the classification of client equipment. And if the record medium concerning claim 9 of this invention is used, the distributed-processing approach concerning above-mentioned claim 5 can be used to the computer used as server equipment or client equipment, and construction of the distributed processing system concerning above-mentioned claim 1 will be attained.

[0245] By moreover, the thing for which it has the common module which does not ask server equipment about an execution environment according to the distributed-processing approach concerning the DPS concerning claim 2 of this invention, and claim 6 Since an activation location can be shifted so that the common module which should be performed in this server equipment may originally be performed in other server equipments to the resource lack condition of server equipment During activation of a service program, for example, by access of many client equipments When the resource of server equipment is pressed, the common module currently performed in server equipment can be performed now in other server equipments which have allowances in a resource, and lowering of the throughput by the lack of a resource of server equipment can be avoided. And if the record medium concerning claim 10 of this invention is used, the distributed-processing approach concerning above-mentioned claim 6 can be used to the computer used as server equipment or client equipment, and construction of the distributed processing system concerning above-mentioned claim 2 will be attained.

[0246] According to the distributed-processing approach concerning the DPS concerning claim 3 of this invention, and claim 7, by moreover, the thing for

which server equipment is equipped with the common module which can be performed in the execution environment of the both sides of server equipment and client equipment Since an activation location can be shifted so that the common module which should be performed in client equipment may originally be performed in server equipment to the resource lack condition of client equipment During activation of a service program, for example, by activation of other application programs etc. When the resource of client equipment is pressed, the common module currently performed in client equipment can be performed now in server equipment, and lowering of the throughput by the lack of a resource of client equipment can be avoided. And if the record medium concerning claim 11 of this invention is used, the distributed-processing approach concerning above-mentioned claim 7 can be used to the computer used as server equipment or client equipment, and construction of the distributed processing system concerning above-mentioned claim 3 will be attained.

[0247] Moreover, according to the distributed-processing approach concerning the distributed processing system concerning claim 4 of this invention, and claim 8 Server equipment can notify renewal of a services module to client equipment.

Further the advice Since it was carried out to the client equipment of the point which distributed the services module in the past In client equipment, it becomes unnecessary for a user to be conscious of the updating stage of the module especially in server equipment, and the update process of an efficient services module of him is attained. And if the record medium concerning claim 12 of this invention is used, the distributed-processing approach concerning above-mentioned claim 8 can be used to the computer used as server equipment or client equipment, and construction of the distributed processing system concerning above-mentioned claim 4 will be attained.

[0248] Moreover, since the initialization module which enables distribution of a different services module for every classification of client equipment is distributed to client equipment according to invention of claim 13 Also when modification of a services module required for the service program for using service at a client equipment side etc. arises It can respond only by changing the content of the initialization module, and management of many services modules which exist for every classification of client equipment does so the effectiveness that easy server equipment is obtained.

[0249] By moreover, the thing for which it has the common module which does

not ask an execution environment according to invention of claim 14 Since an activation location can be shifted so that the common module which should be performed in this server equipment may originally be performed in other server equipments to a resource lack condition When the lack of a resource arises, the common module currently performed can be performed now in other server equipments which have allowances in a resource, and the effectiveness that server equipment with possible this avoiding lowering of a throughput is obtained is done so.

[0250] Moreover, since only the common module by which the lack of a resource is canceled among two or more common modules currently performed is set as the object of an activate request to other server equipments according to invention of claim 15, the effectiveness that the server equipment which can avoid the big pressure to the resource of the server equipment of the activate request point is obtained is done so.

[0251] Moreover, directly since [according to invention of claim 16 / to the client equipment which needs activation of not only the server equipment that emitted the activate request for access to the common module by which the activate request was carried out but its common module] it is possible, the effectiveness

that the server equipment which makes easy utilization of a common module by which the activate request was carried out is obtained is done so.

[0252] Moreover, according to invention of claim 17, since the activation location of a common module is managed on a module activation location table, the effectiveness that the server equipment which makes access of a desired common module easy is obtained by referring to this module activation location table is done so.

[0253] Moreover, since advice of updating is performed only to client equipment [need / the services module updated by referring to a module distribution place table / to be acquired] according to invention of claim 18, the effectiveness that the server equipment which can update an efficient services module is obtained is done so.

[0254] According to invention of claim 19, moreover, only by acquiring and performing an initialization module from server equipment with a service utilization means Since only a services module required to use service is acquirable from server equipment with processing of this initialization module While the service offered by server equipment can use same, a user does so the effectiveness that the client equipment which can use easily the service which

server equipment offers is obtained, without being conscious of the classification of client equipment.

[0255] Moreover, according to invention of claim 20, in a service utilization means, the examination device in which the function and throughput of client equipment are investigated is established. Since the services module which acquires this examination device from server equipment according to the results of an investigation performed and acquired at the time of activation of an initialization module can be chosen When results of an investigation show lowering of a throughput, only the services module which can be performed is acquired and performed. The effectiveness that the client equipment in which the service utilization which becomes possible to request activation of the services module of an and also [it is the need] from server equipment etc., and does not have stress in service activation by this is possible is obtained is done so.

[0256] Moreover, since the services module is acquired from server equipment and it registers with a client data control table when the services module required to use service for a client data control table is not registered according to claim 21, attached storage etc. can be equipped only with a minimum services

module required for utilization of service, and the effectiveness that the client equipment which can realize the effective activity of a resource is obtained is done so.

[0257] Moreover, the resource lack condition in the condition of performing the common module which does not ask an execution environment according to claim 22 is received. Originally the common module which should be performed in this client equipment Since an activation location can be shifted so that it may perform in server equipment, when the lack of a resource arises The common module currently performed can be performed now in the server equipment which has allowances in a resource, and the effectiveness that the client equipment which can avoid lowering of a throughput by this is obtained is done so.

[0258] Moreover, since only the common module by which the lack of a resource is canceled among two or more common modules currently performed is set as the object of an activate request to server equipment according to claim 23, the effectiveness that the client equipment which can avoid the big pressure to the resource of the server equipment of the activate request point is obtained is done so.

[0259] Moreover, according to claim 24, since the activation location of a common module is managed on a module activation location table, access of a desired common module does so the effectiveness that easy client equipment is obtained, by referring to this module activation location table.

[0260] Moreover, since according to claim 25 the updated services module is acquired with an updating module acquisition means when renewal of a services module is notified from server equipment, it becomes unnecessary for the user himself to be conscious of the updating stage of the module in server equipment, and the effectiveness that the client equipment which can update an efficient services module is obtained is done so.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] In the DPS concerning this invention, it is the explanatory view showing the configuration of module of the service data which are offered, and which were prepared for every service.

[Drawing 2] It is the explanatory view showing the system activation gestalt of the server concerning this invention.

[Drawing 3] It is the explanatory view having shown the service module data to each services module of the server data control table concerning this invention.

[Drawing 4] It is the explanatory view showing the system activation gestalt of the client concerning this invention.

[Drawing 5] It is the explanatory view having shown the service module data to

each services module of the client data control table concerning this invention.

[Drawing 6] It is the flow chart which showed the processing performed in starting of the client demon concerning this invention.

[Drawing 7] It is the flow chart which showed the processing performed at the time of starting of the server demon concerning this invention.

[Drawing 8] It is the flow chart which showed processing by the system monitoring thread concerning this invention.

[Drawing 9] It is the flow chart which showed the initialization module acquisition processing concerning this invention.

[Drawing 10] It is the flow chart which showed processing by the initialization module concerning this invention.

[Drawing 11] It is the flow chart which showed the service starting processing concerning this invention.

[Drawing 12] It is the flow chart which showed the module execution environment shift processing in the client concerning this invention.

[Drawing 13] It is the flow chart which showed the services module access processing concerning this invention.

[Drawing 14] It is the flow chart which showed the module execution

environment shift processing in the server concerning this invention.

[Drawing 15] It is the flow chart which showed the services module update process concerning this invention.

[Drawing 16] It is the explanatory view having shown the system configuration at the time of the DPS concerning the gestalt 1 of operation service un-starting.

[Drawing 17] In the gestalt 1 of operation, it is the explanatory view showing service data required in order to perform Service#1.

[Drawing 18] It is the explanatory view having shown the system configuration after service starting of the DPS concerning the gestalt 1 of operation.

[Drawing 19] In the gestalt 1 of operation, it is the explanatory view showing the service module data of initialization module ServiceModule#C1 of Server#1.

[Drawing 20] In the gestalt 1 of operation, it is the explanatory view showing the service module data of common module ServiceModule#D1 of Client#1.

[Drawing 21] It is the explanatory view having shown the system configuration after service starting of the DPS concerning the gestalt 1 of operation.

[Drawing 22] In the gestalt 1 of operation, it is the explanatory view showing the service module data of common module ServiceModule#D1 of Server#1.

[Drawing 23] In the gestalt 1 of operation, it is the explanatory view showing the

service module data of common module ServiceModule#D1 of Client#1.

[Drawing 24] It is the explanatory view having shown the system configuration after service starting of the DPS concerning the gestalt 2 of operation.

[Drawing 25] In the gestalt 2 of operation, it is the explanatory view showing the service module data of common module ServiceModule#D1 of Server#1.

[Drawing 26] In the gestalt 2 of operation, it is the explanatory view showing the service module data of common module ServiceModule#D1 of Client#1.

[Description of Notations]

SD1, SD2 Server demon

CD1 Client demon

SDT1, SDT2 Server data control table

CDT1 Client data control table

STH1, STH2, CTH1 System monitoring thread

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2000-172657

(P2000-172657A)

(43)公開日 平成12年6月23日(2000.6.23)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 15/177	6 7 4	G 0 6 F 15/177	6 7 4 A 5 B 0 4 5
13/00	3 5 7	13/00	3 5 7 Z 5 B 0 8 9
15/16	6 2 0	15/16	6 2 0 B

審査請求 未請求 請求項の数25 O L (全 35 頁)

(21)出願番号 特願平10-349067

(22)出願日 平成10年12月8日(1998.12.8)

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72)発明者 瀬口 義之

福岡県福岡市早良区百道浜2丁目2番1号
富士通九州通信システム株式会社内

(72)発明者 山島 弘之

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74)代理人 100089118

弁理士 酒井 宏明

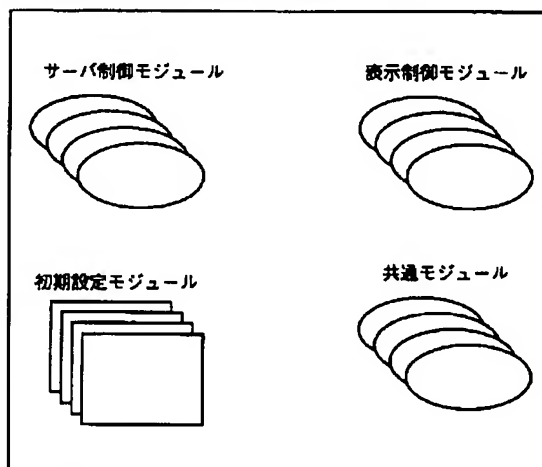
最終頁に続く

(54)【発明の名称】 分散処理システム、分散処理方法、その方法をコンピュータに実行させるプログラムを記録した
コンピュータ読み取り可能な記録媒体、サーバ装置およびクライアント装置

(57)【要約】

【課題】 クライアントの種別の差異にかかわらずサーバからのサービスを快適に利用できる分散処理システムを提供すること。

【解決手段】 サービス実行をおこなうサービスプログラムを、サーバ側でサービス実行をおこなうために必要なサーバ制御モジュール、クライアントに備えられた表示装置に対する表示制御をおこなう表示制御モジュール、クライアント種別毎に用意されかつクライアントに必要なサービスモジュールのサーバからの取得方法を定義する初期設定モジュール、実行位置をクライアントとサーバとに依存しない共通モジュールに分割する。このサービスプログラムの実行において、クライアントのリソース不足が検出された場合は、共通モジュールの一部の起動をサーバに依頼し、サーバのリソース不足が検出された場合は、共通モジュールの一部の起動を他のサーバに依頼する。



【特許請求の範囲】

【請求項1】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおいて、

前記サーバ装置は、サービスの提供および利用をおこなうための個別に実行可能な複数のモジュールと、前記モジュールの一種であり、前記クライアント装置がサービスを利用するのに必要なモジュールを該クライアント装置側で取得するための初期設定モジュールと、前記複数のモジュールを実行することでサービスの提供をおこなうサービス提供手段と、

を備え、

前記クライアント装置は、前記サーバ装置から前記初期設定モジュールを取得し、取得した初期設定モジュールの実行によって前記サーバ装置から前記必要なモジュールを取得し、取得したモジュールを実行することでサービスの利用をおこなうサービス利用手段を備えたことを特徴とする分散処理システム。

【請求項2】 前記サーバ装置は、前記モジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを備え、

前記サービス提供手段は、前記共通モジュールを実行している状態において、前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、同機能を有する他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項1に記載の分散処理システム。

【請求項3】 前記サーバ装置は、前記モジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを備え、

前記サービス利用手段は、前記共通モジュールを実行している状態において、前記クライアント装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項1または2に記載の分散処理システム。

【請求項4】 前記サーバ装置は、前記モジュールが更新された際に、該モジュールを配信した先のクライアント装置に、前記更新されたモジュールの更新を通知する更新通知手段を備え、

前記クライアント装置は、前記更新されたモジュールの更新が通知された場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得手段を備えたことを特徴とする請求項1～3のいずれか一つに記載のサーバ装置。

【請求項5】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、

前記サーバ装置が、前記クライアント装置の要求に応じて、サービスを利用するのに必要なモジュールを該クライアント装置側で取得するための初期設定モジュールを配信する初期設定モジュール配信ステップと、

前記クライアント装置が、前記初期設定モジュールを取得し、取得した初期設定モジュールを実行して、前記サーバ装置から前記必要なモジュールを取得し、取得したモジュールを実行する初期設定モジュール実行ステップと、

を含むことを特徴とする分散処理方法。

【請求項6】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、

前記サーバ装置が、前記サーバ装置と他のサーバ装置との双方の実行環境において実行可能な共通モジュールを実行するサーバ共通モジュール実行ステップと、

前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップと、

を含んだことを特徴とする分散処理方法。

【請求項7】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、

前記クライアント装置が、前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを実行するクライアント共通モジュール実行ステップと、

前記クライアント装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップと、

を含んだことを特徴とする分散処理方法。

【請求項8】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、

前記サーバ装置が、前記クライアント装置の要求に応じて、サービスを利用するのに必要なモジュールを配信した際に、配信先のクライアント装置のアドレスをモジュール配信先テーブルに登録するモジュール配信先登録ステップと、

前記モジュールが更新された際に、前記モジュール配信先テーブルに登録されたクライアント装置のアドレス先に、前記更新されたモジュールの更新を通知するモジュール更新通知ステップと、

前記クライアント装置が、前記更新されたモジュールの更新の通知を受けた場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得ステ

ップと、

を含んだことを特徴とする分散処理方法。

【請求項9】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置から、前記クライアント装置の要求に応じて前記クライアント装置においてサービスを利用するのに必要なモジュールを該クライアント装置側で取得す

るための初期設定モジュールを配信する初期設定モジュール配信ステップと、
前記クライアント装置において前記初期設定モジュールを取得し、取得した初期設定モジュールを実行することで前記サーバ装置から前記必要なモジュールを取得し、取得したモジュールを実行する初期設定モジュール実行

ステップと、
を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項10】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを前記サーバ装置において実行するサーバ共通モジュール実行ステップと、

前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップと、
を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項11】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを前記クライアント装置において実行するクライアント共通モジュール実行ステップと、

前記クライアント装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップと、

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項12】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置から、前記クライアント装置においてサービスを利用するのに必要なモジュールを配信した際に、配信先のクライアント装置のアドレスをモジュール配信先テーブルに登録するモジュール配信先登録ステップと、

前記モジュールが更新された際に、前記モジュール配信先テーブルに登録されたクライアント装置のアドレス先に、前記更新されたモジュールの更新を通知するモジュール更新通知ステップと、

前記クライアント装置に、前記更新されたモジュールの取得の指示が通知された場合に、前記サーバ装置から前記更新されたモジュールを取得させる更新モジュール取得ステップと、

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項13】 ネットワークに接続されてクライアント装置とともに分散処理システムを構築するサーバ装置において、

サービスの提供および利用をおこなうための個別に実行可能な複数のモジュールと、

前記モジュールの一種であり、前記クライアント装置がサービスを利用するのに必要なモジュールを該クライアント装置側で取得するための初期設定モジュールと、

前記クライアント装置からの前記初期設定モジュールの取得要求に応じて該クライアント装置の種別に応じた初期設定モジュールを配信し、該初期設定モジュールが前記クライアント装置において実行されることによって要求されるモジュールを配信し、前記複数のモジュールを実行することでサービスの提供をおこなうサービス提供手段と、

を備えたことを特徴とするサーバ装置。

【請求項14】 前記モジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを備え、

前記サービス提供手段は、前記サーバ装置の処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構を設け、前記共通モジュールを実行している状態において、前記リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、同機能を有する他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項13に記載のサーバ装置。

【請求項15】 前記サービス提供手段は、複数の前記共通モジュールを実行している状態において、前記サー

バ装置のリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、前記他のサーバ装置に対して前記選択した共通モジュールの起動の要求を繰り返すことを特徴とする請求項14に記載のサーバ装置。

【請求項16】 前記サービス提供手段は、前記他のサーバ装置から前記共通モジュールの起動を要求された場合に、前記共通モジュールを起動し、前記他のサーバ装置または前記クライアント装置からの前記起動した共通モジュールへのアクセスを許可することを特徴とする請求項14または15に記載のサーバ装置。

【請求項17】 前記共通モジュールの起動をおこなう前記サーバ装置の実行位置情報を登録するモジュール実行位置テーブルを備え、

前記サービス提供手段は、前記モジュール実行位置テーブルに登録された前記実行位置情報に基づいて前記共通モジュールにアクセスすることを特徴とする請求項14～16のいずれか一つに記載のサーバ装置。

【請求項18】 前記モジュールを配信した先のクライアント装置のアドレスを登録するモジュール配信先テーブルと、

前記サーバ装置において前記モジュールが更新された際に、前記モジュール配信先テーブルに登録されたクライアント装置のアドレス先に前記更新されたモジュールの更新を通知する更新通知手段と、

を備えたことを特徴とする請求項13～17のいずれか一つに記載のサーバ装置。

【請求項19】 ネットワークに接続されてサーバ装置とともに分散処理システムを構築するクライアント装置において、

前記サーバ装置が提供するサービスを利用するのに必要なモジュールを前記サーバ装置から取得するための初期設定モジュールを前記サーバ装置から取得し、取得した初期設定モジュールの実行によって必要なモジュールを取得し、取得したモジュールを実行することでサービスの利用をおこなうサービス利用手段を備えたことを特徴とするクライアント装置。

【請求項20】 前記初期設定モジュールは、前記クライアント装置の処理能力や負荷状態の調査をおこない、該調査結果に応じて、前記サーバ装置が提供するサービスを利用するのに必要なモジュールのうち前記サーバ装置から取得するモジュールを選択することを特徴とする請求項19に記載のクライアント装置。

【請求項21】 前記サーバ装置から取得したモジュールを管理するモジュール管理テーブルを備え、

前記サービス利用手段は、前記モジュール管理テーブルにサービスを利用するのに必要なモジュールが登録されていない場合に、該モジュールを前記サーバ装置から取得し、取得したモジュールを前記モジュール管理テーブルに登録することを特徴とする請求項19または20に

記載のクライアント装置。

【請求項22】 前記サービス利用手段は、サービスを利用するのに必要なモジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを前記サーバ装置から取得し、

前記クライアント装置の処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構を設け、前記共通モジュールを実行している状態において、前記リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項19～21のいずれか一つに記載のクライアント装置。

【請求項23】 前記サービス利用手段は、複数の前記共通モジュールを実行している状態において、前記クライアント装置のリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、前記サーバ装置に対して前記選択した共通モジュールの起動の要求を繰り返すことを特徴とする請求項22に記載のクライアント装置。

【請求項24】 前記共通モジュールの起動をおこなう前記サーバ装置の実行位置情報を登録するモジュール実行位置テーブルを備え、

前記サービス利用手段は、前記モジュール実行位置テーブルに登録された前記実行位置情報に基づいて前記共通モジュールにアクセスすることを特徴とする請求項22または23に記載のクライアント装置。

【請求項25】 前記サーバ装置から更新されたモジュールの更新の通知を受けた場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得手段を備えたことを特徴とする請求項19～24のいずれか一つに記載のクライアント装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワークを介して接続されたサーバおよびクライアントから構成され、ユーザにサービスを提供する分散処理システム、分散処理方法、その方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体、サーバ装置およびクライアント装置において、特に、サービスを実現するために必要なサービスプログラムを実行する際のサーバとクライアントの双方における負荷とサービスプログラムの保守作業とを低減させる分散処理システム、分散処理方法、その方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体、サーバ装置およびクライアント装置に関する。

【0002】

【従来の技術】近年、通信インフラストラクチャの整備

の向上と、LAN (Local Area Network) やインターネットの普及により、ネットワークを介して所望のサービスを受けることを可能とする分散処理システムの利用が一般的になっている。特に、ユーザの利用する複数のクライアント装置 (以下、単にクライアントと称する) と、サービスを提供するサーバ装置 (以下、単にサーバと称する) とから構成され、クライアントがサーバにアクセスすることによって所望のサービスを受けることのできるクライアント・サーバ方式は、ネットワーク上において種々のサービス資源を共有または管理する上での最適なネットワーク接続方式として確立されたものとなっている。

【0003】一方、LANからWAN (Wide Area Network) へ、さらにインターネットへと、地域または社内において点在したネットワーク間を相互接続することによって、ユーザすなわちクライアントが接続できるネットワークの規模が拡大している。さらに、ネットワークを利用するユーザの数、すなわち各ネットワーク内に所属するクライアントの数が増大しており、上記したネットワークの規模の拡大にともなう、サーバの負荷および通信トラフィックの増大が問題となっている。

【0004】このため、現在においては、小規模なネットワーク内においても、提供するサービス種に応じて複数のサーバを用意し、サービス実行における負荷の分散がおこなわれる傾向がある。例えば、データベースサービスを提供するデータベースサーバ、ファイルを一括管理するファイルサーバ、電子メールを管理するメールサーバ、大規模かつ複雑な計算処理を専門とする計算処理サーバ等のサーバがネットワーク上に分散して配置される。

【0005】このようにネットワーク上のサービス授受に関するソフトウェアの進展が見られる一方で、ネットワークに接続される装置、特にクライアントとなりうる多種のハードウェアに対する進展が望まれる。クライアントは、主に、オペレーティングシステムや、種々のアプリケーションプログラムを実行するCPU (Central Processing Unit) と、ネットワークに接続するための通信インターフェースと、CPUによって実行されるプログラムやデータを記憶する記憶装置と、キーボードやマウス等の入力装置と、CRTや液晶ディスプレイ等の表示装置と、から構成される。サーバもまた、クライアントと同様な構成であるが、さらに、各種サービスを提供するための周辺装置が接続される場合がある。

【0006】クライアントの構成は、CPUの種類によってその仕様が太別されており、多くの場合、このCPUに適したオペレーティングシステムを採用することでスタンドアロンとして動作することが可能である。

【0007】したがって、異なるハードウェア仕様のク

ライアント間においても、ネットワークに接続して共有資源の活用を図ろうとする要望が必然と生じてくる。すなわち、どのようなハードウェア仕様のクライアントからであっても、サーバからの同様なサービスを受けたいというユーザの要求が高まってくる。この要求に応えるためには、ネットワークで採用されている通信プロトコルと同一の通信プロトコルを使用した通信をおこなうこと、すなわち通信インターフェースを共通にすることに加えて、サービスを提供するサーバにアクセスするためのサービスプログラムが必要になる。このサービスプログラムは、クライアント上で実行されるため、クライアントのハードウェア仕様、特にオペレーティングシステムに応じて個々に用意する必要がある。

【0008】従来の分散型グループウェアや分散型ネットワークサービスなどは、一般的に、サービスに必要な機能は上記したサービスプログラムとしてあらかじめサーバおよびクライアントに登録されており、このサービスプログラムが、異種のハードウェア仕様または異種のオペレーティングシステム間のインターフェースの違いを吸収して、サーバからクライアントへのサービス提供を実現している。

【0009】以下に、クライアントがサーバからサービスを受ける例として、クライアントがサーバのデータベースから所望のデータを取得する場合について説明する。この場合、アクセス先となるサーバは、大容量のハードディスク等の記憶装置を備え、この記憶装置の記憶領域上にデータベースを構築しているデータベースサーバである。なお、クライアントは、このデータベースサーバにアクセスし、所望のデータを検索して取得するためのサービスプログラム (以下、検索プログラムと称する) をあらかじめ搭載している。

【0010】まず、クライアントは、上記した検索プログラムを起動する。なお、この起動段階において、すでにデータベースサーバとの接続が確立されたものとする。検索プログラムは、一般に、検索条件 (クエリー) の入力を受け付けるクエリー入力処理と、サーバから転送される検索結果を表示する結果表示処理を含んでいる。ユーザは、クライアントに備えられた入力装置を使用して、欲するデータに関するキーワードの入力や検索プログラムによって提示されたカテゴリの選択を進めることによって、クエリーの入力を完了させる。

【0011】検索プログラムは、クエリー入力処理が完了すると、入力されたクエリーの情報をデータベースサーバに送信して、データの検索および転送を要求する。データベースサーバは、クライアントの検索プログラムに対応するサービスプログラムがデーモンとして実行されており、クライアントからのクエリーの情報を受信すると直ちに、このクエリー情報に基づいて、サーバの記憶装置に蓄積された膨大なデータからユーザの欲するデータを特定する。つづいて、データベースサーバは、特

定したデータを、要求のあったクライアントに向けて送信する。

【0012】そして、クライアントの検索プログラムは、データベースサーバから送信されてきたデータを受信し、このデータを表示装置上に表示する。これにより、ユーザは、所望のデータを取得することができる。このように、クライアントは、検索プログラムという比較的軽量のプログラムをサービスプログラムとして搭載することのみで、膨大な容量を必要とするデータベースを利用することができる。

【0013】また、従来における分散処理システムの他の例として、特開平10-21081号公報に開示のクライアント・サーバシステムが提案されている。このクライアント・サーバシステムは、クライアントにおいて実行されるアプリケーションプログラム（サービスプログラム）を、個別に実行可能な複数のモジュールに分割し、この複数のモジュールから構成されるアプリケーションプログラムをサーバの記憶装置上に保存する。そして、クライアントがこのサーバにアクセスしてサービスを利用したい場合には、サーバに保存されたアプリケーションプログラムのうち、実行に必要な部分のモジュールのみをクライアントの記憶装置にダウンロードし、このモジュールを実行する。

【0014】実行中のモジュールにおいて、さらに他のモジュールの実行が必要になる場合には、そのモジュールをサーバからダウンロードする。この際、クライアントにおいてダウンロードされたモジュールの総容量が所定の容量を超える場合には、超過した分のモジュールをクライアントの記憶装置から削除する。これにより、クライアントにおいて、大容量のアプリケーションプログラムを実行する必要がある場合にも、クライアントの記憶装置等の資源を圧迫することがなくなる。

【0015】よって、以上に説明した従来の分散処理システムにおいては、ネットワーク上の共有資源を利用することができるとともに、多種多様な仕様のクライアント間においても、通信インターフェースを共通にすることによりネットワークへの接続が確立でき、サーバが提供するサービスを利用できる。

【0016】

【発明が解決しようとする課題】しかしながら、従来の分散処理システムでは、サーバおよびクライアントに登録されたサービスプログラムは固定的であり、サービスの一部機能の追加や変更をおこなう場合はサーバやクライアントに登録されたサービスプログラム全体の変更をおこなう必要があり、効率的な保守をおこなうことができなかった。

【0017】また、サービスプログラムは一般的にサーバ用プログラムとクライアント用プログラムとして明確に区別されているため、クライアントやサーバの処理能力や機能に応じて、サービス処理の実行位置を移行し負

荷の分散をおこなうことは困難であった。

【0018】また、従来の分散処理システムでは、ユーザが、クライアントに登録するサービスプログラムをクライアントの資源や機能に応じて選択することができるが、これはクライアント毎に実現可能なサービスを制限するものであり、サーバの提供するすべてのサービスを利用できないという、クライアントに依存したサービスの制限の問題があった。

【0019】さらに、従来の分散処理システムでは、一般的に、ユーザまたはシステム管理者が、クライアント毎に、そのクライアントの機種に応じたサービスプログラムをあらかじめインストールしなければならず、ユーザまたはシステム管理者の負担を大きくする傾向があった。

【0020】また、上記したインストールの必要性に関連して、サービスプログラムに対する機能追加や変更が行われる際にも、サーバおよびクライアント装置のサービスプログラム全体を再インストールする必要があり、サービスプログラム内の個別の機能に対する保守は困難であった。さらに、すべてのユーザがサービスプログラムの更新時期を把握することが困難であるという問題があった。

【0021】また、従来の分散型グループウェアや分散型ネットワークサービスでは、一つのクライアント装置で複数のユーザを対象としたマルチユーザシステムを採用している場合が多いが、ユーザ毎に利用するサービスの機能が異なる場合であっても、全ての機能をサービスプログラムとしてクライアントに保持する必要があり、必ずしも効率的にクライアントの資源を利用したものでなかった。

【0022】特開平10-21081号公報に開示のクライアント・サーバシステムは、サービスプログラムをモジュール化して実行に必要なモジュールのみがクライアントにダウンロードされるので、上述したサービスプログラムのインストール作業の問題、サービスプログラム内の個別の機能に対する保守の問題、クライアントの資源の問題を解決するが、サービスプログラムの更新時期を迅速に把握できるものではなかった。

【0023】また、このクライアント・サーバシステムにおいては、クライアントがサービスを受ける際には、サーバからのモジュールのダウンロードを必ず実行する必要がある。したがって、クライアントのCPUがモジュールを実行できる状態までに、少なくともこのダウンロードのための時間が必要とされ、新規なモジュールをつぎつぎと呼び出すようなサービスプログラムを実行する場合には、スループットが低下し、クライアントの負荷が増大するだけでなく、サーバの負荷および通信トラヒックの増大を招来する。

【0024】さらに、このクライアント・サーバシステムにおいては、記憶装置のみに対する資源の有効活用を

可能としており、クライアントが同時に複数のサーバにアクセスする場合等におけるCPUに対する負荷、またはCPUの実行対象位置となる不揮発性メモリの圧迫を考慮したものではなく、必ずしも他のサーバやクライアントを含むネットワーク全体のパフォーマンスを向上させるものではなかった。

【0025】本発明は、上記に鑑みてなされたものであって、クライアントの種別の差異にかかわらずサーバからのサービスを快適に利用できる分散処理システムを提供することを目的とする。

【0026】

【課題を解決するための手段】上述した課題を解決し、目的を達成するために、請求項1の発明において、ネットワークを介して互いに接続されたサーバ装置（サーバ）とクライアント装置（クライアント）とを備えた分散処理システムは、サーバが、サービスの提供および利用をおこなうための個別に実行可能な複数のモジュール（サービスモジュール）と、サービスモジュールの一種であり、クライアントがサービスを利用するのに必要なサービスモジュール（表示制御モジュール等）をそのクライアント側で取得するための初期設定モジュールと、複数のサービスモジュール（サーバ制御モジュール等）を実行することでサービスの提供をおこなうサービス提供手段（後述する「サーバデーモン起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるサーバ側処理に相当）と、を備え、クライアントが、サーバから初期設定モジュールを取得し、取得した初期設定モジュールの実行によってサーバから必要なサービスモジュールを取得し、取得したモジュールを実行することでサービスの利用をおこなうサービス利用手段（後述する「クライアントデーモン起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）を備えたことを特徴とする。

【0027】この請求項1の発明によれば、サーバは、クライアントの要求に応じて、クライアントの種別毎に異なり、かつサービスの実行に必要なモジュールをサーバから取得することのできる初期設定モジュールをクライアントに配信し、クライアントは、サーバからこの初期設定モジュールを取得して実行することで、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができ、この取得したサービスモジュールを実行することにより、サービスを利用することができるので、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、クライアントに応じた初期設定モジュールを取得するだけで、サーバによって提供されるサービスがクライアント間において差異なく利用でき、また、クライアントにおいてサービス利用手段により該当する初期設定モジュールを自動的に取得することができるの

で、ユーザは、クライアントの種別を意識することなく、サーバの提供するサービスを容易に利用することができる。

【0028】また、請求項2の発明において、請求項1に記載の分散処理システムは、サーバが、サービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールを備え、さらに、サービス提供手段が、この共通モジュールを実行している状態（後述する「サービス起動処理」のステップS115s、「クライアントにおけるモジュール実行環境移行処理」のステップS123s、または「サーバにおけるモジュール実行環境移行処理」のステップS143sに相当）において、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（後述する「システム監視スレッドの処理」に相当）に、このサーバと同機能を有する他のサーバに対してその他のサーバ側での共通モジュールの起動を要求する（「サーバにおけるモジュール実行環境移行処理」のステップS145s1に相当）ことを特徴とする。

【0029】この請求項2の発明によれば、サーバに、実行環境を問わない共通モジュールが備えられることで、サーバのリソース不足状態に対して、本来、このサーバにおいて実行すべき共通モジュールを、他のサーバにおいて実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアントのアクセスにより、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるようになり、サーバのリソース不足によるスループットの低下を回避することができる。

【0030】また、請求項3の発明において、請求項1または2に記載の分散処理システムは、サーバが、サービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールを備え、さらに、クライアントのサービス利用手段が、この共通モジュールをクライアント側で実行している状態（「サービス起動処理」のステップS112cに相当）において、クライアントが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、サーバに対してそのサーバ側での共通モジュールの起動を要求する（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）ことを特徴とする。

【0031】この請求項3の発明によれば、サーバにサーバとクライアントとの双方の実行環境において実行可能な共通モジュールが備えられることで、クライアントのリソース不足状態に対して、本来、クライアントにおいて実行すべき共通モジュールを、サーバにおいて実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、他のアプリケー

ションプログラム等の実行により、クライアントの資源が圧迫された場合に、クライアントにおいて実行されていた共通モジュールをサーバにおいて実行できるようになり、クライアントのリソース不足によるスループットの低下を回避することができる。

【0032】また、請求項4の発明において、請求項1～3のいずれか一つに記載の分散処理システムは、サーバが、サービスモジュールの更新が生じた際に、そのサービスモジュールを配信した先のクライアントに、更新されたサービスモジュールの更新を通知する更新通知手段（後述する「サービスモジュール更新処理」のサーバ側処理に相当）を備え、クライアントは、サーバから更新されたサービスモジュールの更新が通知された場合に、更新されたサービスモジュールをサーバから取得する更新モジュール取得手段（「サービスモジュール更新処理」のクライアント側処理に相当）を備えたことを特徴とする。

【0033】この請求項4の発明によれば、サーバが、クライアントに対してサービスモジュールの更新を通知することができ、さらに、その通知は、過去にそのサービスモジュールの配信をおこなった先のクライアントに対しておこなわれるので、クライアントにおいて、ユーザが、特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0034】また、請求項5の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、サーバが、クライアントの要求に応じて、サービスを利用するのに必要なモジュール（サービスモジュールであり、表示制御モジュール等）をそのクライアント側で取得するための初期設定モジュールを配信する初期設定モジュール配信ステップ（「初期設定モジュール取得処理」におけるサーバ側処理に相当）と、クライアントが、この初期設定モジュールを取得し、取得した初期設定モジュールを実行して、サーバから必要なサービスモジュールを取得し、取得したサービスモジュールを実行する初期設定モジュール実行ステップ（「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）と、を含むことを特徴とする。

【0035】この請求項5の発明によれば、クライアントが、サーバから、クライアントの種別毎に異なり、かつサービスの実行に必要なサービスモジュールの取得をおこなう初期設定モジュールを取得して、これを実行することで、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができ、この取得したサービスモジュールの実行により、サービスの利用を可能とするので、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続さ

れた場合であっても、ユーザは、サーバによって提供されるサービスを他のクライアントを利用した場合と同様に利用することができる。

【0036】また、請求項6の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、サーバが、サーバと他のサーバとの双方の実行環境において実行可能な共通モジュールを実行するサーバ共通モジュール実行ステップ（「サービス起動処理」のステップS115s、「クライアントにおけるモジュール実行環境移行処理」のステップS123s、または「サーバにおけるモジュール実行環境移行処理」のステップS143sに相当）と、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、他のサーバに対してその他のサーバ側での共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップ（「サーバにおけるモジュール実行環境移行処理」のステップS145s1に相当）と、を含んだことを特徴とする。

【0037】この請求項6の発明によれば、サーバにおいて、実行環境を問わない共通モジュールの実行中に、サーバのリソース不足状態が生じた際、この共通モジュールを、他のサーバにおいて実行するように要求して、共通モジュールの実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアントのアクセスにより、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるようになり、サーバのリソース不足によるスループットの低下を回避することができる。

【0038】また、請求項7の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、クライアントが、サーバとクライアントとの双方の実行環境において実行可能な共通モジュールを実行するクライアント共通モジュール実行ステップ（「サービス起動処理」のステップS112cに相当）と、クライアントが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、サーバに対してそのサーバ側での共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップ（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）と、を含んだことを特徴とする。

【0039】この請求項7の発明によれば、クライアントにおいて、実行環境を問わない共通モジュールの実行中に、クライアントのリソース不足状態が生じた際、この共通モジュールを、サーバにおいて実行するように要求して、共通モジュールの実行位置を移行することがで

きるので、たとえば、サービスプログラムの実行中に、他のアプリケーションプログラム等の実行により、クライアントの資源が圧迫された場合に、クライアントにおいて実行されていた共通モジュールをサーバにおいて実行できるようになり、クライアントのリソース不足によるスループットの低下を回避することができる。

【0040】また、請求項8の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、サーバが、クライアントの要求に応じて、サービスを利用するのに必要なサービスモジュール（初期設定モジュール、表示制御モジュール等）を配信した際に、配信先のクライアントのアドレスをモジュール配信先テーブル（サーバデータ管理テーブル）に登録するモジュール配信先登録ステップ（「初期設定モジュール取得処理」のステップS93sおよび「初期設定モジュール処理」のステップS102sに相当）と、サービスモジュールが更新された際に、モジュール配信先テーブルに登録されたクライアントのアドレス先に、更新されたサービスモジュールの更新を通知するモジュール更新通知ステップ（「サービスモジュール更新処理」のサーバ側処理に相当）と、クライアントが、更新されたモジュールの更新の通知を受けた場合に、サーバから更新されたモジュールを取得する更新モジュール取得ステップ（「サービスモジュール更新処理」のクライアント側処理に相当）と、を含んだことを特徴とする。

【0041】この請求項8の発明によれば、サーバが、クライアントに対してサービスモジュールの更新を通知することができ、さらに、その通知は、モジュール配信先テーブルを参照することで、過去にそのサービスモジュールの配信をおこなった先のクライアントに対しておこなわれるので、クライアントにおいて、ユーザが、特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0042】また、請求項9の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバから、クライアントの要求に応じてそのクライアントにおいてサービスを利用するのに必要なモジュール（サービスモジュールであり、表示制御モジュール等）をそのクライアント側で取得するための初期設定モジュールを配信する初期設定モジュール配信ステップ（「初期設定モジュール取得処理」におけるサーバ側処理に相当）と、クライアントにおいて初期設定モジュールを取得し、取得した初期設定モジュールを実行することでサーバから必要なサービスモジュールを取得し、取得したモジュールを実行する初期設定モジュール実行ステップ（「初期設定モジュール

取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）と、を実行させることを特徴とする。

【0043】この請求項9の発明によれば、この記録媒体をサーバおよびクライアントにインストールすることによって、クライアントが、サーバから、クライアントの種別毎に異なり、かつサービスの実行に必要なサービスモジュールの取得をおこなう初期設定モジュールを取得し、これを実行することで、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができ、この取得したサービスモジュールの実行により、サービスの利用を可能とするプログラムを実行することができるので、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、ユーザは、サーバによって提供されるサービスを他のクライアントを利用した場合と同様に利用することができる。

【0044】また、請求項10の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバとクライアントとの双方の実行環境において実行可能な共通モジュールをサーバにおいて実行するサーバ共通モジュール実行ステップ（「サービス起動処理」のステップS115s、「クライアントにおけるモジュール実行環境移行処理」のステップS123s、または「サーバにおけるモジュール実行環境移行処理」のステップS143sに相当）と、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、他のサーバに対してその他のサーバ側での共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップ（「サーバにおけるモジュール実行環境移行処理」のステップS145s1に相当）と、を実行させることを特徴とする。

【0045】この請求項10の発明によれば、この記録媒体をサーバおよびクライアントにインストールすることによって、サーバにおいて、実行環境を問わない共通モジュールの実行中に、サーバのリソース不足状態が生じた際、この共通モジュールを、他のサーバにおいて実行するように要求して、共通モジュールの実行位置を移行することが可能なプログラムを実行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアントのアクセスにより、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるようになり、サーバのリソース不足によるスループットの低下を回避することができる。

【0046】また、請求項11の発明において、ネットワークを介して互いに接続されたサーバとクライアント

とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバとクライアントとの双方の実行環境において実行可能な共通モジュールをクライアントにおいて実行するクライアント共通モジュール実行ステップ（「サービス起動処理」のステップS112cに相当）と、クライアントが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、サーバに対してそのサーバ側での共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップ（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）と、を実行させることを特徴とする。

【0047】この請求項11の発明によれば、この記録媒体をサーバおよびクライアントにインストールすることによって、クライアントにおいて、実行環境を問わない共通モジュールの実行中に、クライアントのリソース不足状態が生じた際、この共通モジュールを、サーバにおいて実行するように要求して、共通モジュールの実行位置を移行することが可能なプログラムを実行することができるので、たとえば、サービスプログラムの実行中に、他のアプリケーションプログラム等の実行により、クライアントの資源が圧迫された場合に、クライアントにおいて実行されていた共通モジュールをサーバにおいて実行できるようになり、クライアントのリソース不足によるスループットの低下を回避することができる。

【0048】また、請求項12の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバから、クライアントにおいてサービスを利用するのに必要なサービスモジュール（初期設定モジュール、表示制御モジュール等）を配信した際に、配信先のクライアントのアドレスをモジュール配信先テーブル（サーバデータ管理テーブル）に登録するモジュール配信先登録ステップ（「初期設定モジュール取得処理」のステップS93sおよび「初期設定モジュール処理」のステップS102sに相当）と、サービスモジュールが更新された際に、モジュール配信先テーブルに登録されたクライアントのアドレス先に、更新されたサービスモジュールの更新を通知するモジュール更新通知ステップ（「サービスモジュール更新処理」のサーバ側処理に相当）と、クライアントに、更新されたサービスモジュールの取得の指示が通知された場合に、サーバから更新されたサービスモジュールを取得させる更新モジュール取得ステップ（「サービスモジュール更新処理」のクライアント側処理に相当）と、を実行させることを特徴とする。

【0049】この請求項12の発明によれば、この記録

媒体をサーバおよびクライアントにインストールすることによって、サーバが、クライアントに対してサービスモジュールの更新を通知することができ、さらに、その通知は、モジュール配信先テーブルを参照することで、過去にそのサービスモジュールの配信をおこなった先のクライアントに対しておこなうことが可能なプログラムを実行することができるので、クライアントにおいて、ユーザが、特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0050】また、請求項13において、ネットワークに接続されてクライアント装置（クライアント）とともに分散処理システムを構築するサーバ装置（サーバ）は、サービスの提供および利用をおこなうための個別に実行可能な複数のモジュール（サービスモジュール）と、サービスモジュールの一種であり、クライアントがサービスを利用するのに必要なサービスモジュール（表示制御モジュール等）をそのクライアント側で取得するための初期設定モジュールと、クライアントからの初期設定モジュールの取得要求に応じてそのクライアントの種別に応じた初期設定モジュールを配信し、その初期設定モジュールがクライアントにおいて実行されることによって要求されるサービスモジュールを配信し、サービスモジュールを実行することでサービスの提供をおこなうサービス提供手段（「サーバデーモン起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるサーバ側処理に相当）と、を備えたことを特徴とする。

【0051】この請求項13の発明によれば、サーバは、クライアントの種別毎に異なるサービスモジュールの配信を可能とする初期設定モジュールをクライアントに配信するので、クライアント側でサービスを利用するためのサービスプログラムに必要なサービスモジュールの変更等が生じた場合にも、初期設定モジュールの内容を変更するだけで対応でき、クライアントの種別毎に存在する多くのサービスモジュールの管理が容易になる。

【0052】また、請求項14において、請求項13に記載のサーバ装置（サーバ）は、サービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールを備え、サービス提供手段が、サーバの処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構（「システム監視スレッドの処理」に相当）を設け、共通モジュールを実行している状態において、リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、このサーバと同機能を有する他のサーバに対してその他のサーバ側での共通モジュールの起動を要求する（「サーバにおけるモジュール実行環境移行処理」のステップS145s1に相当）ことを特徴とする。

【0053】この請求項14の発明によれば、サーバに、実行環境を問わない共通モジュールが備えられることで、サーバのリソース不足状態に対して、本来、このサーバにおいて実行すべき共通モジュールを、他のサーバにおいて実行するように実行位置を移行することができるので、サーバのリソース不足が生じた場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるようになり、サーバのスループットの低下を回避することができる。

【0054】また、請求項15において、請求項14に記載のサーバは、サービス提供手段が、複数の共通モジュールを実行している状態において、サーバのリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、他のサーバに対して選択した共通モジュールの起動の要求を繰り返す（「サーバにおけるモジュール実行環境移行処理」に相当）ことを特徴とする。

【0055】この請求項15の発明によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、他のサーバに対する起動要求の対象とするので、起動要求先のサーバのリソースへの大きな圧迫を回避することができる。

【0056】また、請求項16において、請求項14または15に記載のサーバは、サービス提供手段が、他のサーバから共通モジュールの起動を要求された場合に、共通モジュールを起動し、他のサーバまたはクライアントからの起動した共通モジュールへのアクセスを許可する（「クライアントにおけるモジュール実行環境移行処理」および「サーバにおけるモジュール実行環境移行処理」に相当）ことを特徴とする。

【0057】この請求項16の発明によれば、起動要求された共通モジュールへのアクセスを、起動要求を発したサーバだけでなく、その共通モジュールの実行を必要とするクライアントに対しても直接に可能とするので、起動要求された共通モジュールの利用が容易に実現される。

【0058】また、請求項17において、請求項14～16のいずれか一つに記載のサーバは、共通モジュールの起動をおこなうサーバの実行位置情報を登録するモジュール実行位置テーブル（サーバデータ管理テーブル）を備え、サービス提供手段が、モジュール実行位置テーブルに登録された実行位置情報に基づいて共通モジュールにアクセスする（後述する「サービスモジュールアクセス処理」に相当）ことを特徴とする。

【0059】この請求項17の発明によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスが容易になる。

【0060】また、請求項18において、請求項13～17のいずれか一つに記載のサーバは、サービスモジュールを配信した先のクライアントのアドレスを登録するモジュール配信先テーブル（サーバデータ管理テーブル）と、サーバにおいてサービスモジュールが更新された際に、モジュール配信先テーブルに登録されたクライアントのアドレス先に更新されたサービスモジュールの更新を通知する更新通知手段（「サービスモジュール更新処理」のサーバ側処理に相当）と、を備えたことを特徴とする。

【0061】この請求項18の発明によれば、モジュール配信先テーブルを参照することによって、更新されたサービスモジュールの取得が必要なクライアントのみに更新通知をおこなうので、効率的なサービスモジュールの更新処理が可能となる。

【0062】また、請求項19において、ネットワークに接続されてサーバ装置（サーバ）とともに分散処理システムを構築するクライアント装置（クライアント）は、サーバが提供するサービスを利用するのに必要なサービスモジュール（表示制御モジュール等）をサーバから取得するための初期設定モジュールをサーバから取得し、取得した初期設定モジュールの実行によって必要なサービスモジュールを取得し、取得したサービスモジュールを実行することでサービスの利用をおこなうサービス利用手段（「クライアントデーモン起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）を備えたことを特徴とする。

【0063】この請求項19の発明によれば、クライアントは、サービス利用手段によって、サーバから初期設定モジュールを取得して実行するだけで、この初期設定モジュールの処理により、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができるので、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、サーバによって提供されるサービスがクライアント間において差異なく利用できるとともに、ユーザが、クライアントの種別を意識することなく、サーバの提供するサービスを容易に利用することができる。

【0064】また、請求項20において、請求項19に記載のクライアントは、初期設定モジュールが、クライアントの処理能力や負荷状態の調査をおこない（「初期設定モジュール処理」のステップS104cに相当）、その調査結果に応じて、サーバが提供するサービスを利用するのに必要なサービスモジュールのうちサーバから取得するサービスモジュールを選択する（「初期設定モジュール処理」のステップS105cに相当）ことを特徴とする。

【0065】この請求項20の発明によれば、クライアントにおいて実行されるサービス利用手段にクライアン

10

20

30

40

50

トの機能と処理能力を調査する調査機構が設けられ、初期設定モジュールの実行時にこの調査機構を実行して取得される調査結果に応じて、サーバから取得するサービスモジュールを選択することができるので、たとえば、調査結果がクライアントの処理能力の低下を示す場合に、実行可能なサービスモジュールだけを取得して実行し、サービス実行に必要な他のサービスモジュールの実行をサーバ等に依頼することが可能になり、クライアントにおいてストレスのないサービス利用が可能になる。

【0066】また、請求項21において、請求項19または20に記載のクライアントは、サーバから取得したサービスモジュールを管理するモジュール管理テーブル（クライアントデータ管理テーブル）を備え、サービス利用手段が、モジュール管理テーブルにサービスを利用するのに必要なサービスモジュールが登録されていない場合に、そのサービスモジュールをサーバから取得し、取得したサービスモジュールをモジュール管理テーブルに登録する（「初期設定モジュール処理」のステップS107cに相当）ことを特徴とする。

【0067】この請求項21によれば、クライアントデータ管理テーブルに、サービスを利用するのに必要なサービスモジュールが登録されていない場合に、そのサービスモジュールをサーバから取得して、クライアントデータ管理テーブルに登録するので、クライアントは、サービスの利用に必要な最低限のサービスモジュールだけを付属する記憶装置等に備えておくことができ、クライアントの資源の有効な活用を実現することができる。

【0068】また、請求項22において、請求項19～21のいずれか一つに記載のクライアントは、サービス利用手段が、サービスを利用するのに必要なサービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールをサーバから取得し、クライアントの処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構（「システム監視スレッド処理」に相当）を設け、共通モジュールを実行している状態（「サービス起動処理」のステップS112cに相当）において、リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、サーバに対してそのサーバ側での共通モジュールの起動を要求する（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）ことを特徴とする。

【0069】この請求項22によれば、クライアントが実行環境を問わない共通モジュールを実行している状態において、クライアントのリソース不足状態に対して、本来、このクライアントにおいて実行すべき共通モジュールを、サーバにおいて実行するように実行位置を移行することができるので、クライアントのリソース不足が生じた場合に、クライアントにおいて実行されていた共通モジュールを、リソースに余裕のあるサーバにおいて

実行できるようになり、クライアントのスループットの低下を回避することができる。

【0070】また、請求項23において、請求項22に記載のクライアントは、サービス利用手段が、複数の共通モジュールを実行している状態において、クライアントのリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、前記サーバ装置に対して前記選択した共通モジュールの起動の要求を繰り返す（「クライアントにおけるモジュール実行環境移行処理」に相当）ことを特徴とする。

【0071】この請求項23によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、サーバに対する起動要求の対象とするので、起動要求先のサーバのリソースへの大きな圧迫を回避することができる。

【0072】また、請求項24において、請求項22または23に記載のクライアントは、共通モジュールの起動をおこなうサーバの実行位置情報を登録するモジュール実行位置テーブル（クライアントデータ管理テーブル）を備え、サービス利用手段が、モジュール実行位置テーブルに登録された実行位置情報に基づいて共通モジュールにアクセスする（「サービスモジュールアクセス処理」に相当）ことを特徴とする。

【0073】この請求項24によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスが容易になる。

【0074】また、請求項25において、請求項19～24のいずれか一つに記載のクライアントは、サーバから更新されたサービスモジュールの更新の通知を受けた場合に、サーバから更新されたサービスモジュールを取得する更新モジュール取得手段（「サービスモジュール更新処理」のクライアント側処理に相当）を備えたことを特徴とする。

【0075】この請求項25によれば、更新モジュール取得手段によって、サーバからサービスモジュールの更新が通知された際に、更新されたサービスモジュールを取得するので、クライアントにおいて、ユーザ自らが、サーバにおけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0076】

【発明の実施の形態】以下に、本発明にかかる分散処理システム、分散処理方法、サーバ装置およびクライアント装置の実施の形態を図面に基づいて詳細に説明する。なお、この実施の形態によりこの発明が限定されるものではない。ここでは、まず、本発明にかかる分散処理システム、分散処理方法、サーバ装置およびクライアント装置において動作される各処理についての原理的な説明

をおこない、つづいてそれら各処理によって動作する具体的な実施の形態を説明する。

【0077】本発明にかかる分散処理システムは、LANケーブル、電話回線もしくは専用回線等の通信回線網、すなわちネットワークを介して接続される一つ以上のサーバと一つ以上のクライアントとから構成され、クライアントが、主にサーバの提供するサービスをネットワーク上の共有資源として利用できるシステムである。特に、複数のサーバがネットワークに接続されることにより、サービス実行における負荷が分散され、クライアントの受けるサービスがストレスなく実行できる環境が提供される。

【0078】また、この分散処理システムは、通信インターフェースを共通にすることで、ハードウェア仕様やオペレーティングシステムの異なる複数種のクライアントが接続されてもよい。

【0079】ここにおけるサービスは、クライアントの要求に対して、サーバがクライアントに種々のデータを提供することであり、データベースサービスや計算処理サービス等である。よって、サーバは、クライアントからの要求に従って目的とするサービスを実行するためのサーバ用のサービスプログラムを備え、クライアントもまた、サーバに対してサービスを要求するためのクライアント用のサービスプログラムを備える必要がある。

【0080】クライアントは、オペレーティングシステム、上記したクライアント用のサービスプログラム、その他のアプリケーションプログラムを実行するCPUと、ネットワークに接続するための通信インターフェースと、CPUによって実行されるプログラムやデータを記憶する記憶装置と、キーボードやマウス等の入力装置と、CRTや液晶ディスプレイ等の表示装置と、から構成され、通常のコンピュータシステムを用いることができる。また、サーバは、クライアントと同様な構成であるが、さらに、複数のCPUを備えてマルチプロセッサ対応の高速な処理を可能とすることや、大容量の記憶装置、各種サービスを提供するための周辺装置が接続される場合がある。なお、上記した記憶装置は、RAM、ROM等のメモリやハードディスク装置、さらにICカードメモリ、フロッピーディスク、光磁気ディスク、CD-ROM等の取り外し可能な記録媒体を装填可能とするドライブ装置を含む。

【0081】本発明にかかる分散処理システムでは、上記したサービスプログラムを個別に実行可能な複数のモジュールから構成し、サーバ用サービスプログラムとクライアント用サービスプログラムはともに、モジュール単位で実行可能である。また、クライアント用サービスプログラムを構成するモジュール群は、サービスデータとしてサーバからクライアントへ配信することが可能であり、これらモジュール（以下、サービスモジュールと称する）の保守・管理は主にサーバによっておこなわれ

る。

【0082】図1は、提供されるサービス毎に用意されたサービスデータのモジュール構成を示す説明図である。図1において、サービスデータには、サーバ側でサービス実行をおこなうために必要なサーバ制御モジュール、クライアントに備えられた表示装置に対する表示制御をおこない、グラフィカルユーザインターフェース（GUI）およびその他の入出力制御をおこなう表示制御モジュール、クライアント種別毎に用意され、クライアントに必要なサービスモジュールのサーバからの取得方法を定義しかつクライアントの処理能力と負荷状態の調査をおこなう機能を有する初期設定モジュール、さらにサービスモジュールの実行位置をクライアント、サーバに依存しない共通モジュールに分割される。

【0083】上述したサーバ用サービスプログラムおよびクライアント用サービスプログラムは、図1に示すサービスデータにおいて、必要とするサービスモジュールを組合せることによって構成され、また各サービスモジュール（サーバ制御モジュール、表示制御モジュール、初期設定モジュール、共通モジュール）を複数備えることができる。

【0084】たとえば、サーバ用サービスプログラムは、いくつかのサーバ制御モジュールから構成され、クライアント用サービスプログラムは、一つの初期設定モジュールと、いくつかの表示制御モジュールと、いくつかの共通モジュールとから構成される。

【0085】図2は、サーバのシステム実行形態を示す説明図である。サーバは、オペレーティングシステムSOSの実行環境下において、複数のクライアントからの接続要求の受け付け、サービスモジュールの配送、サーバのサービス制御をおこなうサーバデーモンSD1が常駐している。この状態において、クライアントからのサービス実行要求が発生すると、サーバデーモンSD1中に、該当するサービスプログラムSSP1（Service-1）のサービスモジュールを実行するためのスレッドを生成し、このサービスモジュールを実行する。

【0086】また、サーバデーモンSD1には、サーバの処理能力と負荷状態を逐次管理するシステム監視スレッドSTH1が含まれており、一定時間間隔でサーバに対するシステム監視がおこなわれる。また、サーバは、記憶装置に、サーバおよびクライアントに登録されるサービスモジュールの情報を管理するサーバデータ管理テーブルSDT1を備えており、必要に応じてこのサーバデータ管理テーブルSDT1を参照し、サーバデーモンSD1内のスレッドとのデータの入出力をおこなう。

【0087】なお、サーバデーモンSD1は、オペレーティングシステムSOSに対してデータおよび命令の入出力をおこない、オペレーティングシステムSOSは、通信用のソケットAPI（Application Program Interface）を介してネットワ

ークとのデータの入出力、すなわちクライアントまたは他のサーバとの接続を確立している。

【0088】サーバデータ管理テーブルSDT1は、サーバの提供するサービスのサービス名と、このサービスを実行するのに必要なサービスモジュールのサービスモジュール名と、各サービスモジュールの版数および実行状態に関する情報を含むサービスモジュールデータと、から構成される。図3は、サーバデータ管理テーブルSDT1の各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【0089】図3において、サーバデータ管理テーブルSDT1におけるサービスモジュールデータは、たとえば、対応する「サービスモジュール名」、対応するサービス名を示す「デフォルトサービス名」、サーバがサービスモジュールを配送したクライアントを示す「配送先クライアントアドレスリスト」、サービスモジュールの種別（サーバ制御モジュール、表示制御モジュール、初期設定モジュール、共通モジュール）を示す「モジュール種別」、クライアントのオペレーティングシステムまたはハードウェア仕様を示す「プラットフォーム種別」、サービスモジュールの「版数」、サービスモジュールの実行位置を示す「状態フラグ」、「状態フラグ」がローカル実行を示す場合のサービスモジュールに対するアクセスコードを示す「ローカル実行時のアクセスID」、「状態フラグ」が「リモート実行」を示す場合のサービスモジュールの属するサーバのアドレスを示す「リモート実行時のサーバアドレス」、「状態フラグ」が「リモート実行」を示す場合のサービスモジュールに対するアクセスコードを示す「アクセスID」等のデータから構成される。

【0090】図4は、クライアントのシステム実行形態を示す説明図である。クライアントは、オペレーティングシステムCOSの実行環境下において、サーバとの通信をおこなうクライアントデーモンCD1が常駐している。ここで、オペレーティングシステムCOSは、サーバのオペレーティングシステムSOSと必ずしも同じである必要はない。

【0091】この状態において、ユーザからのサービス実行要求が発生すると、クライアントデーモンCD1中に、該当するサービスプログラムCSP1（Service1）のサービスモジュールを実行するためのスレッドを生成し、このサービスモジュールを実行する。

【0092】また、クライアントデーモンCD1には、クライアントの処理能力と負荷状態を逐次管理するシステム監視スレッドCTH1が含まれており、一定時間間隔でクライアントに対するシステム監視がおこなわれる。また、クライアントは、記憶装置に、サーバから取得してクライアントに登録されるサービスモジュールの情報を管理するクライアントデータ管理テーブルCDT1を備えており、必要に応じてこのクライアントデータ

管理テーブルCDT1を参照し、クライアントデーモンCD1内のスレッドとのデータの入出力をおこなう。

【0093】なお、クライアントデーモンCD1は、オペレーティングシステムCOSに対してデータおよび命令の入出力をおこない、オペレーティングシステムCOSは、サーバと同様に、通信用のソケットAPIを介してネットワークとのデータの入出力、すなわちサーバとの接続を確立している。

【0094】クライアントデータ管理テーブルCDT1は、クライアントに登録されたサービスのサービス名と、このサービスを実行するのに必要なサービスモジュールのサービスモジュール名と、各サービスモジュールの版数および実行状態に関する情報を含むサービスモジュールデータと、から構成される。図5は、クライアントデータ管理テーブルCDT1の各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【0095】図5において、クライアントデータ管理テーブルCDT1におけるサービスモジュールデータは、たとえば、図3に説明した「サービスモジュール名」、「デフォルトサービス名」、「版数」、「状態フラグ」、「ローカル実行時のアクセスID」、「リモート実行時のサーバアドレス」、「アクセスID」と、サービスモジュールの取得先であるサーバのアドレスを示す「取得サーバアドレス」等のデータから構成される。

【0096】つぎに、図2および図4に示したサーバおよびクライアントの実行形態において、サービスが実行されるまでの流れをフローチャートにしたがって説明する。

【0097】（クライアントデーモンの起動処理）まず、クライアントとなる端末装置は、多くの場合、起動と同時に、オペレーティングシステムCOSの環境下において、クライアントデーモンCD1が実行される。これにより、クライアントデーモンCD1は、常にクライアントデーモンとして生成されたスレッドが実行されており、各スレッドの実行内容にしたがったデータの入出力がおこなわれる。

【0098】図6は、クライアントデーモンCD1の起動において実行される処理を示したフローチャートである。図6において、たとえば、オペレーティングシステムCOSの起動と同時に自動実行されるタスク起動プログラムによって、まず、クライアントデーモンCD1が起動される（ステップS61）。クライアントデーモンCD1はサーバからのメッセージを受信したり、クライアントで実行されるサービスモジュールからのメッセージ要求をサーバに送信する。

【0099】クライアントで実行されるサービスモジュールには、ユーザが入力装置を用いることでおこなわれるボタン押下やテキスト入力などのサービス選択やサービス実行に関するイベントを制御する表示制御モジュール

ル、クライアント種別に応じたサービスモジュールの取得を可能にする初期設定モジュール、クライアントかサーバかの実行位置に依存しない共通モジュールが含まれる。

【0100】このクライアントデーモンCD1の起動にともなって、クライアントの処理能力や負荷状態の監視をおこなうシステム監視スレッドCTH1が生成されて実行される(ステップS62)。このようにシステム監視スレッドCTH1が常駐した後は、現在サーバに登録されているサービスのリスト、すなわちサーバが提供可能なサービスのリストを取得する(ステップS63)。

【0101】実際には、クライアントデーモンCD1内に、サーバからのサービスリスト取得をおこなうためのスレッドが生成され、このスレッドがオペレーティングシステムCOSに対してサービスリストの取得を要求する。オペレーティングシステムCOS(より詳しくは通信ソケットAPI)は、クライアントがアクセス可能なサーバに対してサービスリストの配信を要求する。

【0102】このように、サービスリストの取得をおこなうスレッドもまた、サービスモジュールとして機能し、クライアントにおいて最低限必要なモジュールとなる。よって、ユーザは、クライアントデーモンCD1の起動時のみでなく、必要に応じてサーバからサービスリストの取得をおこなうことができる。

【0103】サーバからサービスリストが配信されると、上記したサービスリスト取得スレッドまたはあらかじめ登録された表示制御モジュールのスレッドが、クライアントの表示装置上にサービスリストにしたがったGUI表示をおこない、ユーザによる入力装置を介したアクセスもしくはサーバアクセスの待ち受け状態に移移する(ステップS64)。

【0104】ここで、サーバアクセスとは、サーバがクライアントに対しておこなうメッセージ送信やクライアント上で実行されているサービスモジュールに対する命令またはデータの送信を示す。

【0105】(サーバデーモンの起動処理)図6に示したステップS64の後の処理は後述するとして、つぎに、サーバデーモンSD1の起動処理について説明する。サーバとなる端末装置は、オペレーティングシステムSOSの環境下において、通常、サーバデーモンSD1が常駐状態であり、サーバデーモンとして生成されたスレッドが常に実行されている。サーバは、クライアントからのアクセスを常に受け付ける必要があるため、保守作業等の特別な場合を除いては、常に起動状態であると考えてよい。このため、サーバデーモンSD1の起動処理もまた稀にしかおこなわれないが、ここでは、サーバの起動時またはシステム管理者の操作によるサーバデーモンSD1の起動がおこなわれる場合を説明する。

【0106】図7は、サーバデーモンSD1の起動時に実行される処理を示したフローチャートである。図7に

おいて、まず、サーバデーモンSD1が起動される(ステップS71)。サーバデーモンSD1は、クライアントや他のサーバからのメッセージを受信したり、サーバ内で実行されるサービスモジュールからのメッセージ要求をクライアントや他のサーバに送信する。

【0107】サーバで実行されるサービスモジュールには、常駐型のサービスやクライアントからのサービス起動要求により起動されるサーバ制御モジュール、クライアントかサーバかの実行位置に依存しない共通モジュールが含まれる。

【0108】このサーバデーモンSD1の起動にともなって、サーバの処理能力や負荷状態の管理をおこなうシステム監視スレッドSTH1が生成されて実行される(ステップS72)。そして、システム監視スレッドSTH1が常駐した後は、クライアントからのアクセスもしくは他サーバからのアクセス待ち受け状態に移移する(ステップS73)。

【0109】(システム監視スレッドの処理)つぎに、上述した「クライアントデーモンCD1の起動処理」におけるシステム監視スレッドCTH1の実行(ステップS62)および「サーバデーモンの起動処理」におけるシステム監視スレッドSTH1の実行処理(ステップS72)について説明する。なお、システム監視スレッドCTH1およびSTH1は、ともに同一の流れの処理を示す。

【0110】図8は、システム監視スレッドによる処理を示したフローチャートである。システム監視スレッドは、クライアントおよびサーバにそれぞれ配備され、クライアントデーモンCD1およびサーバデーモンSD1によって生成される。つづいて、システム監視スレッドは、このシステム監視スレッドが常駐しているシステム(サーバまたはクライアント)の処理能力や負荷状態のチェックをおこなう(ステップS81)。ここで、処理能力のチェックとしては、メモリや記憶装置のデータ蓄積容量、サーバやクライアントとの通信に必要な通信ソケットの空き数、スレッド生成の許容量等の調査があげられ、負荷状態のチェックとしては、CPU稼働率やベンチマークテストの実施による計測があげられる。

【0111】つづいて、ステップS81の処理能力・負荷状態のチェックの結果、システムが、処理能力の低下や著しい負荷状態、すなわちリソース不足の状態を示すかどうか判定される(ステップS82)。ステップS82において、システムのリソース不足の状態が示された場合には、さらに、このシステム監視スレッドの実行位置がクライアントであるかどうか判定される(ステップS84)。

【0112】ステップS84において、システム監視スレッドの実行位置がクライアントであれば、後述するクライアント用のモジュール実行環境移行処理(ステップS120)が実行され、システム監視スレッドの実行位

置がサーバであれば、後述するサーバ用のモジュール実行環境移行処理（ステップS140）が実行される。

【0113】一方、ステップS82において、リソース不足が検出されない場合には、一定時間の待ち受け状態（ステップS83）となり、この一定時間経過後に再びステップS81の処理がおこなわれる。すなわち、図8に示すシステム監視スレッドの処理は、一定時間毎にシステムの処理能力・負荷状態のチェックをおこない、そのチェックの結果がリソース不足を示す場合には、このシステム監視スレッドが実行されている位置（サーバまたはクライアント）に応じたモジュール実行環境移行処理が実行されることを示す。

【0114】（初期設定モジュール取得処理）つぎに、図6に示したステップS64のユーザアクセス／サーバアクセス待ち受け状態に対して、ユーザによるサービス選択がおこなわれ、その結果実行される初期設定モジュール取得処理について説明する。図9は、初期設定モジュール取得処理を示したフローチャートである。

【0115】図9において、まず、ユーザが、クライアントの表示装置上にGUI表示されたサービスリストから、マウスやキーボード等の入力装置を用いて、所望のサービスを選択する（ステップS91c）。そして、選択されたサービスを実行するのに必要な初期設定モジュールを、上記したクライアントデータ管理テーブルCDT1から検索する（ステップS92c）。

【0116】つづいて、該当する初期設定モジュールが、クライアントデータ管理テーブルCDT1に登録されているかどうかの判定がおこなわれる（ステップS93c）。ステップS93cにおいて、該当する初期設定モジュールがクライアントデータ管理テーブルCDT1に登録されている場合は、サーバに登録されている初期設定モジュールの版数をチェックするために、サーバに登録された初期設定モジュールのうち、上記した初期設定モジュールに対応する初期設定モジュールの情報を取得する（ステップS94c）。

【0117】ステップS94cの処理の手順としては、該当する初期設定モジュールを表すサービスモジュール名をともなった初期設定モジュールの情報取得要求をサーバに対して発信する。そして、クライアントは、この初期設定モジュール情報取得要求に対してサーバから配信される返答、すなわち該当する初期設定モジュールの情報を受け取る。

【0118】クライアントが受け取った初期設定モジュール情報は、たとえば図3に示すサービスモジュールデータのように構成されており、このデータに含まれる版数情報と、クライアントに登録された初期設定モジュール情報、たとえば図5に示すサービスモジュールデータに含まれる版数情報と、の比較をおこない、版数が一致するかどうかを判定する（ステップS95c）。

【0119】ステップS95cにおいて、版数が一致し

ない場合、すなわちクライアントに登録された初期設定モジュールの版数が、サーバに登録された初期設定モジュールの版数よりも古い値を示す場合には、クライアントにおいて該当する初期設定モジュールを変更する必要があるため、サーバに対して初期設定モジュールの取得要求を発信する。

【0120】また、ステップS93cにおいて、該当する初期設定モジュールがクライアントデータ管理テーブルCDT1に登録されていない場合もまた、サーバに対して初期設定モジュールの取得要求を発信する。なお、初期設定モジュール取得要求には、目的とする初期設定モジュールに対応したサービス名と、クライアントのハードウェア仕様またはオペレーティングシステムの種類を特定するプラットフォーム種別（クライアント種別）とを特定する数値情報が含まれている。

【0121】サーバは、クライアントから受信した初期設定モジュール取得要求の示す引数（上記した数値情報）によって、サービス名とクライアント種別を特定する（ステップS91s）。つづいて、サーバは、サーバデータ管理テーブルSTD1から、ステップS91sにおいて特定されたサービス名とクライアント種別に対応した初期設定モジュールを選択する（ステップS92s）。

【0122】そして、サーバは、ステップS92sにおいて選択された初期設定モジュールを、初期設定モジュール取得要求のあったクライアントに対して配信する。サーバは、初期設定モジュールの配信後、サーバデータ管理テーブルSDT1において、配信した初期設定モジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、初期設定モジュール取得要求のあったクライアントのアドレスを登録する（ステップS93s）。

【0123】一方、初期設定モジュールを受信したクライアント（より詳しくは、クライアントデモンCD1）は、クライアントデータ管理テーブルCDT1に、受信した初期設定モジュールの登録をおこなう。この際、取得先のサーバのアドレスを、図5に示すサービスモジュールデータ内の「取得サーバアドレス」に登録する（ステップS96c）。

【0124】つづいて、クライアントは、ステップS96cにおいて登録された初期設定モジュールのスレッドを生成し（ステップS97c）、この初期設定モジュールを起動する（ステップS98c）。

【0125】なお、以上に説明した初期設定モジュール取得処理は、上述したサービスリスト取得スレッドまたはこのスレッドにより構成されるサービスプログラムに含めてもよいし、初期設定モジュール取得処理をおこなうためのサービスモジュールをクライアントにあらかじめ登録しておくことで実行されるようにしてもよい。

【0126】（初期設定モジュール処理）つぎに、図9

に示したステップS98cにつづく処理、すなわち初期設定モジュールの処理について説明する。図10は、初期設定モジュールによる処理を示したフローチャートである。図10において、まず、初期設定モジュールは、その内部に定義された情報から、サービスの実行に必要なサービスモジュールを抽出する(ステップS101c)。

【0127】そして、初期設定モジュールは、クライアントデータ管理テーブルCDT1から、ステップS101cにおいて抽出したサービスモジュールの検索をおこなう(ステップS102c)。ステップS102cにおいて、クライアントデータ管理テーブルCDT1から検索対象となるサービスモジュールを特定することができる場合、すなわち検索対象となるサービスモジュールがクライアントデータ管理テーブルCDT1に登録されている場合には、そのサービスモジュールの版数をチェックする(ステップS103c)。

【0128】ステップS103cにおけるサービスモジュールの版数のチェックは、まず、サーバに登録されているサービスモジュールの版数を取得するために、サーバに登録されたサービスモジュールのうち、対応するサービスモジュールの情報の取得要求をそのサーバに対して発信する必要がある。

【0129】クライアントは、このサービスモジュール情報取得要求の発信後、サーバから配信される返答、すなわち該当するサービスモジュール情報を受け取り、初期設定モジュールは、受け取ったサービスモジュール情報から版数情報を取得する。なお、このサービスモジュール情報取得要求をおこなうにおいて、複数のサービスモジュールの情報が必要である場合は、該当する複数のサービスモジュールの情報を一度の送受信によって取得することも可能である。

【0130】一方、クライアントに登録されたサービスモジュールに対しては、クライアントデータ管理テーブルCDT1において、対象となるサービスモジュールに対応するサービスモジュールデータから、版数情報を取得することができる。なお、ステップS102cにおいて、検索対象となるサービスモジュールがクライアントデータ管理テーブルCDT1に登録されていない場合には、ステップS103cの処理はスキップされ、つぎのステップS104cの処理が実行される。

【0131】ステップS104cにおいては、オペレーティングシステムCOSに対するシステムコールによって、現在のクライアントの機能、処理能力および負荷状態のチェックがおこなわれる。このクライアントに対するシステムチェックは、上述したシステム監視スレッドCTH1によっておこなうこともでき、その場合、初期設定モジュールは、システム監視スレッドCTH1に対してシステムがリソース不足状態にあるかどうかの情報の取得要求をおこなうことができるが、ここでは、特

に、システムの迅速な状態チェックをおこなうために、システムコールを用いた初期設定モジュール独自のシステムチェックをおこなうものとする。

【0132】つづいて、初期設定モジュールは、ステップS101cにおいて抽出されたサービスモジュールをサーバから取得する必要があるかどうかの判定をおこなう(ステップS105c)。ステップS105cにおいて、サービスモジュールをサーバから取得する必要がある場合、具体的には、ステップS102cにおいてサービスモジュールがクライアントデータ管理テーブルCDT1に登録されていない場合や、ステップS103cにおいて取得した両者のサービスモジュールの版数情報が異なる場合に、サーバに対してサービスモジュールの取得要求を発信する。

【0133】サーバは、クライアントから受信したサービスモジュール取得要求に含まれる情報によって、サービスモジュールを特定し、サーバデータ管理テーブルSDT1から、特定されたサービスモジュールを選択する(ステップS101s)。

【0134】そして、サーバは、ステップS101sにおいて選択されたサービスモジュールを、サービスモジュール取得要求のあったクライアントに対して配信する。サーバは、サービスモジュールの配信後、サーバデータ管理テーブルSDT1において、配信したサービスモジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、サービスモジュール取得要求のあったクライアントのアドレスを登録する(ステップS102s)。

【0135】一方、サービスモジュールを受信したクライアント(より詳しくは、クライアントデーモンCD1)は、まず、受信したサービスモジュールをクライアントデータ管理テーブルCDT1に登録するかどうかの判定をおこなう(ステップS106c)。ステップS106cの処理は、ステップS104cにおけるシステムチェック、またはシステム監視スレッドCTH1において、リソース不足状態が示された場合に、実行すべきサービスモジュール、特に共通モジュールの実行をサーバに依頼するための処理を示している。

【0136】ステップS106cにおいて、サービスモジュールの登録をおこなう場合、すなわちクライアントのリソース不足が検出されない場合には、クライアントデータ管理テーブルCDT1に、サーバから受信したサービスモジュールの登録をおこなう。この際、取得先のサーバのアドレスを、図5に示すサービスモジュールデータ内の「取得サーバアドレス」に登録する(ステップS107c)。

【0137】ステップS106cにおいて、サービスモジュールの登録をおこなわない場合、すなわちクライアントのリソース不足が検出された場合には、後述するクライアント用モジュール実行環境移行処理(図8のステ

10

20

30

40

50

ップS120)がおこなわれる。

【0138】そして、クライアント用モジュール実行環境移行処理がおこなわれた後、またはステップS107cの終了後に、以下に説明するサービスの起動処理がおこなわれる。

【0139】(サービス起動処理) つぎに、上述した図9のステップS91cにおいて選択されたサービスの起動処理について説明する。図11は、サービス起動処理を示したフローチャートである。図11において、まず、クライアントデーモンCD1は、ユーザからのサービス起動要求により、サーバに対してサーバ制御をおこなうサーバ制御モジュールの起動要求をおこなう。この要求によって、サーバは、該当するサーバ制御モジュールを起動する(ステップS111s)。

【0140】このとき、クライアント処理能力の低下が著しく、さらに過負荷状態の場合、すなわち、上述した「初期設定モジュール処理」のステップS106cにおいて、サービスモジュールの登録をおこなわない場合、またはシステム監視スレッドCTH1によって新たにリソース不足が検出された場合には、クライアントデーモンCD1は、本来自端末で実行すべきである共通モジュールの起動をサーバに依頼するために共通モジュール起動要求をおこなう。

【0141】よって、ステップS111sの処理につづく共通モジュール起動要求の有無の判定(ステップS112s)が必要となる。ステップS112sにおいて、共通モジュール起動要求が存在する場合には、まず、このサーバにおける処理能力・負荷状態のチェックがおこなわれる(ステップS113s)。すなわち、サーバデーモンSD1が、システム監視スレッドSTH1から、サーバのリソース状態を示す情報を取得する。

【0142】つづいて、ステップS113sによって取得された情報によって、システムがリソース不足であるかどうかの判定がおこなわれる(ステップS114s)。ステップS114sにおいてリソース不足が示された場合には、起動要求のあった共通モジュールの実行を、さらに他のサーバに依頼するためのサーバ用モジュール実行環境移行処理(ステップS140)が実行される。

【0143】ステップS114sにおいてリソース不足が検出されない場合には、サーバデーモンSD1は、起動を要求された共通モジュールの起動をおこない(ステップS115s)、その共通モジュールへのアクセスに必要な実行アドレス、アクセスID、状態フラグ(図3に示すサービスモジュールデータの「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」参照)を設定する(ステップS116s)。

【0144】そして、起動要求のあったクライアントに対して、サービスの起動、すなわちサーバ制御モジュールと共通モジュールの起動要求があった場合の共通モジ

ュールとの起動の完了を示し、かつステップS116sにおいて設定された実行アドレス、アクセスID、状態フラグの情報を含んだサービス起動完了通知を配信する。その後、サーバデーモンSD1は、クライアントアクセス待ち受け状態に遷移する(ステップS117s)。これは、サーバ用のサービスプログラム(サーバ制御モジュール、共通モジュール)の起動が完了した状態を示す。

【0145】クライアントデーモンCD1は、サーバからサービス起動完了通知を受信すると、実行するサービスに対応した表示制御モジュールを起動する(ステップS111c)。つづいて、実行すべき共通モジュールのうち、システム監視スレッドCTH1等によって取得されるクライアントのシステムリソース情報に応じて、実行できるだけの共通モジュールを起動する(ステップS112c)。

【0146】そして、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1に、サービス実行に必要な共通モジュールのサービスモジュールデータのそれぞれにおいて、「ローカル実行時のサーバアドレス」、「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」等の登録をおこなう(ステップS113c)。これにより、クライアント用のサービスプログラム(表示制御モジュール、共通モジュール)の起動が完了したことが示される。

【0147】よって、ユーザは、このクライアント用のサービスプログラムによってサーバへのアクセス(詳しくは、サーバ用のサービスプログラムへのアクセス)をおこなうことができ、サーバの提供するサービスを受けることができる。

【0148】(クライアントにおけるモジュール実行環境移行処理) つぎに、クライアントにおいて、システム監視スレッドCTH1によるシステムの処理能力の低下や過負荷状態が検出された場合に、共通モジュールの実行環境をサーバに移行する処理について説明する。すなわち、この処理は、図8に示すシステム監視スレッドの処理において、ステップS120のクライアント用モジュール実行環境移行処理に該当する。

【0149】図12は、クライアントにおけるモジュール実行環境移行処理を示したフローチャートである。なお、このモジュール実行環境移行処理は、クライアントにおいてあらかじめ登録されたモジュール実行環境移行プログラムにより提供され、そのスレッドは、クライアントデーモンCD1内に生成されることにより、システムに常駐して実行されるものとする。

【0150】図12において、まず、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1から現在実行中の共通モジュールのうち、一つ目の共通モジュールを抽出する(ステップS121c)。そして、クライアントデーモンCD1は、ステップS121

cにおいて、共通モジュールの抽出が可能であったかどうかを判定する(ステップS122c)。

【0151】ステップS122cにおいて共通モジュールの抽出が可能でなかった場合、すなわち実行に必要な共通モジュールがクライアントデータ管理テーブルCDT1にこれ以上登録されていなかった場合には、このモジュール実行環境移行処理を終了する。また、ステップS122cにおいて共通モジュールの抽出が可能であった場合、抽出された共通モジュールのサービスモジュールデータから状態フラグの情報を取得し、この状態フラグが「ローカル実行」を示すかどうかを判定する(ステップS123c)。

【0152】ステップS123cにおいて状態フラグが「ローカル実行」を示さない場合、すなわち抽出された共通モジュールがサーバにおいて実行されている場合は、実行環境を移行する必要がないので、クライアントデータ管理テーブルCDT1においてつぎの共通モジュールの検索をおこなうために、再びステップS121cの処理に戻る。

【0153】ステップS123cにおいて状態フラグが「ローカル実行」を示す場合、すなわち抽出された共通モジュールがクライアントにおいて実行されている場合は、サーバに対してこの共通モジュールの実行を依頼するための共通モジュール起動要求をおこなう(ステップS124c)。

【0154】サーバは、クライアントからの共通モジュール起動要求を受信すると、まず、このサーバにおける処理能力・負荷状態のチェックをおこなう(ステップS121s)。すなわち、サーバデーモンSD1が、システム監視スレッドSTH1から、サーバのリソース状態を示す情報を取得する。

【0155】つづいて、ステップS121sにおいて取得された情報によって、システムがリソース不足であるかどうかの判定がおこなわれる(ステップS122s)。ステップS122sにおいてリソース不足が検出された場合には、起動要求のあった共通モジュールの実行を、さらに他のサーバに依頼するためのサーバ用モジュール実行環境移行処理(ステップS140)が実行される。

【0156】ステップS122sにおいてリソース不足が検出されない場合には、サーバデーモンSD1は、起動要求された共通モジュールの起動をおこない(ステップS123s)、その共通モジュールへのアクセスに必要な実行アドレス、アクセスID、状態フラグ(図3に示すサービスモジュールデータの「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」参照)を設定する(ステップS124s)。

【0157】そして、起動要求のあったクライアントに対して、共通モジュールの起動の完了を示しかつステップS124sにおいて設定された実行アドレス、アクセ

スID、状態フラグの情報を含んだ共通モジュール起動完了通知を配信する。

【0158】クライアントデーモンCD1は、サーバから共通モジュール起動完了通知を受信すると、ローカル実行中の共通モジュールの実行を停止し、蓄積領域が不足している場合は共通モジュールプログラムを削除する。つづいて、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1において、対象となる共通モジュールのサービスモジュールデータの状態フラグを「リモート実行」に変更する(ステップS125c)。

【0159】そして、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1に、サービス実行に必要な共通モジュールのサービスモジュールデータのそれぞれにおいて、「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」の登録をおこなう(ステップS126c)。

【0160】さらに、クライアントデーモンCD1は、システム監視スレッドCTH1からシステムリソース情報を取得することにより、システムがリソース不足(処理能力の低下または過負荷状態)であるかどうかを判定する(ステップS127c)。ステップS127cにおいてリソース不足が検出されない場合は、このモジュール実行環境移行処理を終了する。

【0161】ステップS127cにおいてリソース不足が検出された場合は、再びステップS121cの処理を繰り返す。すなわち、このモジュール実行環境移行処理は、システムのリソース不足が解消されるまで繰り返される。また、本発明にかかる分散処理システムでは、共通モジュールは他のサービスモジュールやクライアントデーモンCD1によって共有されるため、共通モジュールの実行位置が変更となった場合であっても同様に共通モジュールへのアクセスを保証する必要がある。

【0162】図13は、サービスモジュールアクセス処理を示したフローチャートである。クライアントデーモンCD1または他のサービスモジュールは、実行に必要なサービスモジュールへのアクセスに先だって、図13に示すサービスモジュールアクセス処理をおこなう。図13において、まず、クライアントデーモンCD1または他のサービスモジュールは、クライアントデータ管理テーブルCDT1から、該当するサービスモジュール(ここでは、特に共通モジュール)のサービスモジュールデータを検索する(ステップS131c)。なお、このサービスモジュールアクセス処理を実行するまでに、上述した初期設定モジュールによるサービスモジュールのセットアップは完了しているものとする。

【0163】つづいて、クライアントデーモンCD1または他のサービスモジュールは、ステップS131cにおいて検索されたサービスモジュールデータから状態フラグの情報を取得し、状態フラグが「ローカル実行」を

示すかを判定する(ステップS132c)。ステップS132cにおいて状態フラグが「ローカル実行」でない場合、すなわち状態フラグが「リモート実行」を示す場合には、サービスモジュールデータに含まれた「リモート実行時のサーバアドレス」の示すサーバに対し、このサービスモジュールに対する処理を要求する。なお、この要求の通知には、該当するサービスモジュールにアクセスするための「アクセスID」の情報も含まれている。

【0164】サーバデーモンSD1は、クライアントからのサービスモジュール処理要求を受信して、この要求から処理対象となる「アクセスID」を特定し、この「アクセスID」を用いることで、該当するサービスモジュールのアクセスを達成する(ステップS131s)。

【0165】そして、ステップS132cにおいて状態フラグが「ローカル実行」を示す場合には、該当するサービスモジュールのサービスモジュールデータから「ローカル実行時のアクセスID」を取得し、このアクセスIDを用いることで、該当するサービスモジュールのアクセスを達成する(ステップS133c)。

【0166】よって、このサービスモジュールアクセス処理によれば、共通モジュールの実行位置がクライアントであってもサーバであっても他のサービスモジュールやクライアントデーモンCD1は影響を受けることはない。また、共通モジュールの実行位置は「リモート実行時のサーバアドレス」と「アクセスID」によって規定されるため、リモート実行をおこなうサーバはクライアントが共通モジュールを取得したサーバに限定されるものではない。

【0167】(サーバにおけるモジュール実行環境移行処理) つぎに、サーバにおいて、システム監視スレッドSTH1によってシステムの処理能力の低下や過負荷状態が検出された場合に、共通モジュールの実行環境を他のサーバに移行する処理について説明する。すなわち、この処理は、図8に示すシステム監視スレッドの処理において、ステップS140のサーバ用モジュール実行環境移行処理に該当する。

【0168】図14は、サーバにおけるモジュール実行環境移行処理を示したフローチャートである。なお、このモジュール実行環境移行処理は、サーバにおいてあらかじめ登録されたモジュール実行環境移行プログラムにより提供され、そのスレッドは、サーバデーモンSD1内に生成されることにより、システムに常駐して実行されるものとする。

【0169】図14において、まず、サーバデーモンSD1は、このサーバに対してサービスモジュールの起動要求が発生しているかどうかを判定する(ステップS141s1)。ステップS141s1においてサービスモジュールの起動要求が発生していない場合は、サーバデー

ータ管理テーブルSDT1から現在実行中のサービスモジュールのうち、一つ目のサービスモジュールを抽出する(ステップS142s1)。そして、サーバデーモンSD1は、ステップS142s1において、サービスモジュールの抽出が可能であったかどうかを判定する(ステップS143s1)。

【0170】ステップS143s1においてサービスモジュールの抽出が可能でなかった場合、すなわち実行に必要なサービスモジュールがサーバデータ管理テーブルSDT1にこれ以上登録されていなかった場合には、このモジュール実行環境移行処理を終了する。また、ステップS143s1においてサービスモジュールの抽出が可能であった場合、抽出されたサービスモジュールのサービスモジュールデータから状態フラグの情報を取得し、この状態フラグが「ローカル実行」を示すかどうかを判定する(ステップS144s1)。

【0171】ステップS144s1において状態フラグが「ローカル実行」を示さない場合、すなわち抽出されたサービスモジュールが他のサーバにおいて実行されている場合は、実行環境を移行する必要がないので、サーバデータ管理テーブルSDT1においてつぎのサービスモジュールの検索をおこなうために、再びステップS142s1の処理に戻る。

【0172】ステップS144s1において状態フラグが「ローカル実行」を示す場合、すなわち抽出されたサービスモジュールがこのサーバ(第1のサーバ)において実行されている場合は、他のサーバ(第2のサーバ)に対してこのサービスモジュールの実行を依頼するためのサービスモジュール起動要求をおこなう(ステップS145s1)。

【0173】他のサーバ(第2のサーバ)は、上記したサーバからサービスモジュール起動要求を受信すると、まず、このサーバにおける処理能力・負荷状態のチェックをおこなう(ステップS141s2)。すなわち、第2のサーバのサーバデーモンが、第2のサーバのシステム監視スレッドから、第2のサーバのリソース状態を示す情報を取得する。

【0174】つづいて、ステップS141s2において取得された情報によって、システムがリソース不足であるかどうかの判定がおこなわれる(ステップS142s2)。ステップS142s2においてリソース不足が検出された場合には、起動要求のあったサービスモジュールの実行を、さらに他のサーバ(第3のサーバ)に依頼するためのサーバ用モジュール実行環境移行処理、すなわちこのモジュール実行環境移行処理が、第3のサーバに対して再帰的に実行される。

【0175】ステップS142s2においてリソース不足が検出されない場合には、第2のサーバのサーバデーモンは、起動要求されたサービスモジュールの起動をおこなう(ステップS143s2)、そのサービスモジュ

ールへのアクセスに必要な実行アドレス、アクセスID、状態フラグ（図3に示すサービスモジュールデータの「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」参照）を設定する（ステップS144s2）。

【0176】そして、起動要求を発した第1のサーバに対して、サービスモジュールの起動の完了を示しかつステップS144s2において設定された実行アドレス、アクセスID、状態フラグの情報を含んだサービスモジュール起動完了通知を配信する。

【0177】第1のサーバのサーバデーモンSD1は、第2のサーバまたは他のサーバからサービスモジュール起動完了通知を受信すると、ローカル実行中のサービスモジュールの実行を停止する。つづいて、第1のサーバのサーバデーモンSD1は、サーバデータ管理テーブルSDT1において、対象となるサービスモジュールのサービスモジュールデータの状態フラグを「リモート実行」に変更する（ステップS146s1）。

【0178】そして、第1のサーバのサーバデーモンSD1は、サーバデータ管理テーブルSDT1に、サービス実行に必要なサービスモジュールのサービスモジュールデータのそれぞれにおいて、「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」の登録をおこなう（ステップS147s1）。

【0179】さらに、第1のサーバのサーバデーモンSD1は、システム監視スレッドSTH1からシステムリソース情報を取得することにより、システムがリソース不足（処理能力の低下または過負荷状態）であるかどうかを判定する（ステップS148s1）。ステップS148s1においてリソース不足が検出されない場合は、このモジュール実行環境移行処理を終了する。

【0180】ステップS148s1においてリソース不足が検出された場合は、再びステップS142s1の処理を繰り返す。すなわち、このモジュール実行環境移行処理は、システムのリソース不足が解消されるまで繰り返される。以上に説明したサーバにおけるモジュール実行環境移行処理によってサービスモジュールの実行位置が変更となった場合であっても、図13に示すサービスモジュールアクセス処理によって、サービスモジュールへのアクセスを保証することができる。

【0181】（サービスモジュール更新処理）つぎに、サーバに登録されているサービスモジュールが更新された場合に、サーバデーモンSD1が自動的にクライアントに対してサービスモジュールの更新登録を要求する処理について説明する。

【0182】図15は、サービスモジュール更新処理を示したフローチャートである。図15において、まず、システム管理者等によって、サーバに登録されたサービスモジュールの更新がおこなわれる（ステップS151s）。そして、システム管理者等による指示またはサー

バデーモンSD1によって、サーバデータ管理テーブルSDT1において、更新対象となったサービスモジュールのサービスモジュールデータから、配送先クライアントアドレスリストを取得する（ステップS152s）。

【0183】つづいてサーバデーモンSD1は、ステップS152sにおいて取得した配送先クライアントアドレスリストに示されるクライアントに対して、サービスモジュールの更新の通知となるサービスモジュール更新要求を配信する（ステップS153s）。このサービスモジュール更新要求には、配信対象となるクライアントに応じた初期設定モジュールの版数情報が含まれている。

【0184】ここで、サーバデーモンSD1は、配信対象となるクライアントがネットワークにログオンしているかどうかのチェックをおこない（ステップS154s）、ネットワーク障害や端末未起動によって、クライアントのログオン状態が確認できない場合は、そのクライアントに対するサービスモジュール更新要求の配信をエラー終了としてスキップし、配信対象となる他のクライアントに対してのログオンチェックを繰り返す。なお、サーバデーモンSD1は、このエラー終了によってサービスモジュール更新要求が配信されなかったクライアントに対して、ある一定時間毎にログオンチェックをおこない、サービスモジュール更新要求の配信をこころみる。

【0185】一方、サービスモジュール更新要求を受信したクライアントデーモンCD1は、クライアント管理データテーブルCDT1において初期設定モジュールのサービスモジュールデータから、版数情報を取得し、サービスモジュール更新要求に含まれる初期設定モジュールの版数情報との比較をおこない、初期設定モジュールの変更をおこなうべきかを判定する（ステップS151c）。

【0186】ステップS151cにおいて、初期設定モジュールの変更をおこなう必要がある場合、すなわち、クライアント管理データテーブルCDT1に登録された初期設定モジュールの版数よりサービスモジュール更新要求に含まれる初期設定モジュールの版数が新しい場合は、サーバに対して初期設定モジュールの取得要求を発信する。なお、初期設定モジュール取得要求には、上述した「初期設定モジュール取得処理」と同様に、目的とする初期設定モジュールに対応したサービス名と、クライアントのハードウェア仕様またはオペレーティングシステムの種類を特定するプラットフォーム種別（クライアント種別）とを特定する数値情報が含まれている。

【0187】サーバは、クライアントから受信した初期設定モジュール取得要求の示す数値情報によって、サービス名とクライアント種別を特定し、サーバデータ管理テーブルSDT1から、特定されたサービス名とクライアント種別に対応した初期設定モジュールを選択する。

つづいて、サーバは、この選択された初期設定モジュールを、初期設定モジュール取得要求のあったクライアントに対して配信する。また、サーバは、初期設定モジュールの配信後、サーバデータ管理テーブルSDT1において、配信した初期設定モジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、初期設定モジュール取得要求のあったクライアントのアドレスを登録する。

【0188】クライアントデーモンCD1は、サーバから初期設定モジュールを取得後、この初期設定モジュールをクライアントデータ管理テーブルCDT1に登録する。そして、クライアントデーモンCD1は、上記したクライアントデータ管理テーブルCDT1への初期設定モジュールの登録後、またはステップS151cにおいて初期設定モジュールの変更を必要としない場合に、該当する初期設定モジュールを実行する（ステップS152c）。

【0189】つづいて、クライアントデーモンCD1は、初期設定モジュールに定義されたサービスモジュールをサーバから取得する必要があるかどうかの判定をおこなう（ステップS153c）。ステップS153cにおいて、サービスモジュールをサーバから取得する必要がある場合、具体的には、該当するサービスモジュールがクライアントデータ管理テーブルCDT1に登録されていない場合や、登録されたサービスモジュールの版数情報が古い場合に、サーバに対してサービスモジュールの取得要求を発信する。

【0190】サーバは、クライアントから受信したサービスモジュール取得要求に含まれる情報によって、サービスモジュールを特定し、サーバデータ管理テーブルSDT1から、特定されたサービスモジュールを選択する。そして、サーバは、選択されたサービスモジュールを、サービスモジュール取得要求のあったクライアントに対して配信する。また、サーバは、サービスモジュールの配信後、サーバデータ管理テーブルSDT1において、配信したサービスモジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、サービスモジュール取得要求のあったクライアントのアドレスを登録する。

【0191】一方、サービスモジュールを受信したクライアントは、クライアントデータ管理テーブルCDT1に、サーバから受信したサービスモジュールの登録をおこなう（ステップS154c）。この際、取得先のサーバのアドレスを、図5に示すサービスモジュールデータ内の「取得サーバアドレス」に登録する。よって、ユーザは、このサービスモジュール更新処理により、サービスモジュールの更新を意識することなしに常に最新のサービスモジュールの版数によるサービスを享受することが可能になる。

【0192】なお、以上に説明したサービスモジュール

更新処理は、たとえば、サービスモジュール更新処理をおこなうためのサービスモジュールをクライアントにあらかじめ登録しておくことで実行される。

【0193】つぎに、以上に説明した処理に基づいて動作する本発明の分散処理システムについて、より理解を深めるために、さらに具体的な実施の形態について説明する。

【0194】（実施の形態1）図16は、実施の形態1にかかる分散処理システムのサービス未起動時のシステム構成を示した説明図である。図16において、実施の形態1にかかる分散処理システムは、サーバとしてサービス提供をおこなうServer#1とクライアントとしてサービスを受けるClient#1が、ネットワークを介し、かつ通信インターフェースを共通にして接続されることで構成される。

【0195】なお、実際には、本発明にかかる分散処理システムは、多数のサーバとクライアントがネットワークに接続されて構成されるが、ここでは、理解を容易にするため、一つのサーバと一つのクライアント間における処理を中心に説明する。

【0196】ここで、Server#1のネットワーク上の論理アドレスを#0001とし、Client#1のネットワーク上の論理アドレスを#0011とする。Server#1は、クライアントからの接続要求を受け付け、サービスの実行制御をおこなうサーバデーモンSD1と、サービスデータの管理をおこなうサーバデータ管理テーブルSDT1と、を備えている。また、Server#1は、図7に示したサーバデーモン起動処理がおこなわれた後の状態であり、クライアントアクセス/他のサーバアクセスの待ち受け状態にある。よって、サーバデーモンSD1内には、クライアントの処理能力や負荷状態の管理をおこなうシステム監視スレッドSTH1が生成されている。

【0197】Client#1は、Server#1との通信をおこない、ユーザにインタフェースを提供するクライアントデーモンCD1と、サーバから取得したサービスデータの管理をおこなうクライアントデータ管理テーブルCDT1と、を備えている。また、Client#1は、図6に示したクライアントデーモン起動処理がおこなわれた後の状態であり、ユーザアクセス/サーバアクセスの待ち受け状態にある。よって、クライアントデーモンCD1内には、クライアントの処理能力や負荷状態の管理をおこなうシステム監視スレッドCTH1が生成されている。

【0198】また、Server#1とClient#1はともに、オペレーティングシステムOSの実行環境下において動作され、通信ソケットAPIを介してネットワークとのデータ入出力をおこなう。ここで、上述したように、Server#1とClient#1は、通信インターフェースを共通にしているので、必ずしもオ

ペレーティングシステムOSが同一である必要はない。

【0199】ここで、サーバによって提供されるサービスの一つとしてService #1が実行される場合を考える。図17は、Service #1を実行するために必要なサービスデータを示す説明図である。すなわち、サーバ用サービスプログラムおよびクライアント用サービスプログラムは、このサービスデータに含まれるサービスモジュールの組合せによって構成される。

【0200】図17において、サービスデータService #1は、複数のサービスモジュールから構成され、具体的には、サーバで実行されサービス制御をおこなうサーバ制御モジュールServiceModule #A1~An、クライアントで実行されGUI表示やサーバとの通信をおこなう表示制御モジュールServiceModule #B1~Bn、クライアントで実行されサービス実行に必要なモジュールの取得方法やシステムの処理能力および負荷状態の調査が定義された初期設定モジュールServiceModule #C1~Cn、実行環境をサーバかクライアントかに依存しない共通モジュールServiceModule #D1~Dnから構成される。また、それぞれのサービスモジュールはクライアント種別毎やサービスに依存して必要な数だけ用意される。

【0201】まず、図16に示すようなServer #1とClient #1のサービス未起動状態において、ユーザが、図6に示したクライアントデーモン起動処理によってClient #1の表示装置上に提示されたサービスリストから、所望のサービスを選択する。このサービス選択により、図9に示した初期設定モジュール取得処理が実行され、Server #1に対する初期設定モジュールの取得要求がおこなわれる。なお、ここでは、Client #1のクライアントデータ管理テーブルCDT1に、該当するサービスの実行に必要な初期設定モジュール、表示制御モジュールおよび共通モジュールが登録されていないものとする。

【0202】そして、Server #1は、Client #1のクライアント種別に応じた初期設定モジュールServiceModule #C1を、Client #1に返送する。一方、Client #1は、Server #1から初期設定モジュールServiceModule #C1を受け取り、初期設定モジュールServiceModule #C1の起動をおこなう。この初期設定モジュールServiceModule #C1の起動によって、図10に示した初期設定モジュール処理が実行され、初期設定モジュールServiceModule #C1内の定義内容から、Service #1の実行に必要な表示制御モジュールServiceModule #B1および共通モジュールServiceModule #D1の取得がおこなわれる。

【0203】ここで、Client #1において、Se

ervice #1を実行するには、さらに共通モジュールServiceModule #D2が必要であったが、システム監視スレッドCTH1によりClient #1のリソース不足が検出されたために、共通モジュールServiceModule #D2の取得および起動はおこなわれず、共通モジュールServiceModule #D2の起動をServer #1に依頼するものとする。

【0204】Client #1は、Server #1に対してService #1の起動要求をおこない、図11に示したサービス起動処理が実行される。この際、Server #1は、Client #1から起動要求のあった共通モジュールServiceModule #D2を起動して、クライアントアクセス待ち受け状態に移移する。

【0205】つづいて、Client #1において、表示制御モジュールServiceModule #B1、共通モジュールServiceModule #D1の起動がおこなわれ、Service #1の起動が完了する。

【0206】図18は、実施の形態1にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。図18において、Server #1は、図16に示す状態に加えて、Service #1を提供するサーバ用サービスプログラムとして、サーバ制御モジュールServiceModule #A1、共通モジュールServiceModule #D2が用意され、これらサービスモジュールの実行においてサーバデーモンSD1内にスレッドが生成される。

【0207】また、Client #1は、図16に示す状態に加えて、Service #1を提供するクライアント用サービスプログラムとして、表示制御モジュールServiceModule #B1、初期設定モジュールServiceModule #C1、共通モジュールServiceModule #D1が用意され、これらサービスモジュールの実行においてクライアントデーモンCD1内にスレッドが生成される。

【0208】つぎに、図9に示した初期設定モジュール取得処理において、Server #1がクライアントに初期設定モジュールの配信をおこなった場合のServer #1のサービスモジュールデータについて説明する。ここでは特に、Server #1が複数のクライアント(Client #1、Client #2、Client #3)に対して、初期設定モジュールServiceModule #C1を配信した場合を考える。

【0209】図19は、この場合のServer #1のサーバデータ管理テーブルSDT1に登録された初期設定モジュールServiceModule #C1のサービスモジュールデータを示す説明図である。図19において、まず、「サービスモジュール名」として「Ser

viceModule#C1」が登録され、「デフォルトサービス名」として「Service#1」が登録されている。そして、「配送先クライアントリスト」には、上記した複数のクライアントClient#1、Client#2、Client#3のネットワーク上における論理アドレス（#0011、#0012、#0013）が登録されている。

【0210】また、「モジュール種別」として初期設定モジュールを示すコードである「3」が登録され、「プラットフォーム種別」として、Client#1のハードウェア仕様またはオペレーティングシステムを特定するコード「1」が登録されている。

【0211】さらに、「版数」として「1」が登録されており、初期設定モジュールServiceModule#C1が更新された場合には、この「版数」が更新されて、図15に示したサービスモジュール更新処理がおこなわれる。ここでは、初期設定モジュールServiceModule#C1に対するサービスモジュール更新処理は、「配送先クライアントリスト」に登録されているClient#1、Client#2、Client#3に対して、ユーザが介することのない自動的な処理としておこなわれる。

【0212】つぎに、Client#1のクライアントデータ管理テーブルCDT1に登録されるサービスモジュールデータの例として、共通モジュールServiceModule#D1のサービスモジュールデータについて説明する。

【0213】図20は、この場合のClient#1のクライアントデータ管理テーブルCDT1に登録された共通モジュールServiceModule#D1のサービスモジュールデータを示す説明図である。図20において、まず、「サービスモジュール名」として「ServiceModule#D1」が登録され、「デフォルトサービス名」として「Service#1」が登録されている。そして、「版数」として「1」が登録され、「取得サーバアドレス」としてこの共通モジュールServiceModule#D1を配信したサーバの論理アドレスを示す「#0001」が登録されている。

【0214】さらに、「状態フラグ」としてローカル実行、すなわちClient#1での共通モジュールServiceModule#D1の実行を示すコードである「1」が登録され、「ローカル実行時のアクセスID」としてこの共通モジュールServiceModule#D1にアクセスするために必要なアクセスID「111」が登録されている。

【0215】つぎに、図8に示したシステム監視スレッド処理により処理能力の低下もしくは過負荷状態が検出され、図13に示したサービスモジュールアクセス処理がおこなわれる場合のServer#1およびClient#1のサービスモジュールデータについて説明す

る。ここでは特に、図18に示した状態から、Client#1において過負荷状態が検出され、Client#1がServer#1に対して共通モジュールServiceModule#D1の実行依頼をおこなった場合を考える。

【0216】図21は、実施の形態1にかかる分散処理システムのサービス起動後のシステム構成を示した説明図であり、特に図18に示した状態において、Server#1がClient#1から依頼された共通モジュールServiceModule#D1を起動した状態を示す。よって、図21に示すように、Client#1は、共通モジュールServiceModule#D1をクライアント用のサービスプログラムから開放し、Server#1が共通モジュールServiceModule#D1を起動する。

【0217】Server#1において共通モジュールServiceModule#D1が起動されると、サーバデータ管理テーブルSDT1の共通モジュールServiceModule#D1のサービスモジュールデータが変更される。図22は、この場合のServer#1のサーバデータ管理テーブルSDT1に登録された共通モジュールServiceModule#D1のサービスモジュールデータを示す説明図である。

【0218】図22において、まず、「サービスモジュール名」として「ServiceModule#D1」が登録され、「デフォルトサービス名」として「Service#1」が登録されている。そして、「配送先クライアントリスト」には、上記したクライアントClient#1の論理アドレスである「#0011」が登録されている。

【0219】また、「モジュール種別」として共通モジュールを示すコードである「4」が登録され、「プラットフォーム種別」として、サーバ、クライアント、ハードウェア仕様またはオペレーティングシステムを特定しないコード「0」が登録されている。

【0220】さらに、「版数」として「1」が登録され、「状態フラグ」としてローカル実行、すなわちServer#1での共通モジュールServiceModule#D1の実行を示すコードである「1」が登録されている。そして、「ローカル実行時のアクセスID」としてこの共通モジュールServiceModule#D1にアクセスするために必要なID「222」が登録されている。

【0221】一方、Client#1においてもまた、Server#1におけるモジュール実行が完了すると、クライアントデータ管理テーブルCDT1内の共通モジュールServiceModule#D1のサービスモジュールデータが変更される。図23は、この場合のClient#1のクライアントデータ管理テーブルCDT1に登録された共通モジュールServiceM

odule # D1のサービスモジュールデータを示す説明図である。

【0222】図23において、まず、「サービスモジュール名」として「ServiceModule # D1」が登録され、「デフォルトサービス名」として「Service # 1」が登録されている。そして、「版数」として「1」が登録され、「取得サーバアドレス」としてこの共通モジュールServiceModule # D1を配信したサーバの論理アドレスを示す「#0001」が登録されている。

【0223】さらに、「状態フラグ」としてリモート実行、すなわちサーバによる共通モジュールServiceModule # D1の実行を示すコードである「2」が登録されている。そして、「リモート実行時のサーバアドレス」としてこの共通モジュールServiceModule # D1の起動をおこなうサーバであるServer # 1のアドレスを示す「#0001」が登録され、「アクセスID」としてこのServer # 1上の共通モジュールServiceModule # D1にアクセスするのに必要なID「222」が登録されている。

【0224】以上に説明したように、実施の形態1にかかる分散処理システムによれば、上述した「クライアントデーモンの起動処理」、「サーバデーモンの起動処理」、「初期設定モジュール取得処理」、「初期設定モジュール処理」、「サービス起動処理」の実行によって、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、クライアントに応じた初期設定モジュールを取得するだけで、サーバによって提供されるサービスがクライアント間において差異なく利用でき、また、クライアントのサービス利用手段によって、該当する初期設定モジュールを自動的に取得することができるので、サービスに必要なモジュール構成の知識をユーザに要求することなく、サーバの提供するサービスを容易に利用することが可能となる。

【0225】また、上述した「システム監視スレッドの処理」、「クライアントにおけるモジュール実行環境移行処理」の実行によって、クライアントの処理能力の低下が示される場合に、クライアントにおいて実行されていた共通モジュールの一部をサーバ上で起動することができ、クライアントのリソース不足によるスループットの低下を迅速に回避することができるとともに、効果的な負荷分散を実現することができる。

【0226】さらに、上述した「サービスモジュール更新処理」の実行によって、クライアントを利用するユーザは特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、クライアントにおいて迅速なモジュールの更新処理が可能となる。

【0227】（実施の形態2）つぎに、実施の形態1に

において説明した図19および図20のサービスモジュールデータの登録内容に対応する状態、すなわちClient # 1がServer # 1に対して、共通モジュールServiceModule # D2の起動要求をおこなった状態（図18に示した状態）において、Server # 1のリソース不足により、さらに、他のサーバであるServer # 2に対して共通モジュールServiceModule # D2の起動要求がおこなわれる場合について説明する。

10 【0228】図24は、実施の形態2にかかる分散処理システムのサービス起動時のシステム構成を示した説明図である。図24において、実施の形態2にかかる分散処理システムは、サーバとしてService # 1の提供をおこなうことのできるServer # 1およびServer # 2と、クライアントとしてService # 1を受けるClient # 1とが、ネットワークを介し、かつ通信インターフェースを共通にして接続されることで構成されている。

20 【0229】なお、実際には、本発明にかかる分散処理システムは、多数のサーバとクライアントがネットワークに接続されて構成されるが、ここでは、理解を容易にするため、二つのサーバと一つのクライアント間における処理を中心に説明する。

30 【0230】ここで、Server # 1、Server # 2のネットワーク上の論理アドレスをそれぞれ#0001、#0002とし、Client # 1のネットワーク上の論理アドレスを#0011とする。Server # 1およびServer # 2はともに、図11に示したサービス起動処理を実行可能とする状態にあり、特に、Server # 1は、図8に示したシステム監視スレッド処理によるリソース不足の検出によって、図14に示したモジュール実行環境移行処理が実行された状態にある。

40 【0231】ここでは、図14に示したモジュール実行環境移行処理における他のサーバが、Server # 2に該当し、このServer # 2によって、Client # 1から起動要求のあった共通モジュールServiceModule # D2の起動がおこなわれる。よって、図24に示すように、Server # 2においては、Service # 1を提供するサーバ用サービスプログラムとして、共通モジュールServiceModule # D2が用意され、このサービスモジュールの実行においてサーバデーモンSD2内にスレッドが生成された状態にある。

50 【0232】一方、Client # 1は、実施の形態1において説明した図18に示す状態と同様に、Service # 1を提供するクライアント用サービスプログラムとして、表示制御モジュールServiceModule # B1、初期設定モジュールServiceModule # C1、共通モジュールServiceModu

le#D1が用意され、これらサービスモジュールの実行においてクライアントデーモンCD1内にスレッドが生成された状態にある。

【0233】また、Server#1、Server#2、Client#1は、オペレーティングシステムOSの実行環境下において動作され、通信ソケットAPIを介してネットワークとのデータ入出力をおこなう。ここで、上述したように、Server#1、Server#2およびClient#1は、通信インターフェースを共通にしているので、必ずしもオペレーティングシステムOSが同一である必要はない。

【0234】つぎに、図24に示した状態におけるServer#1およびClient#1のサービスモジュールデータについて説明する。ここでは特に、Client#1がServer#1に対して共通モジュールServiceModule#D2の実行依頼をおこない、さらにServer#1がServer#2に対して共通モジュールServiceModule#D2の実行依頼をおこなった場合を考える。

【0235】図25は、この場合のServer#1のサーバデータ管理テーブルSDT1に登録された共通モジュールServiceModule#D2のサービスモジュールデータを示す説明図である。

【0236】図25において、まず、「サービスモジュール名」として「ServiceModule#D2」が登録され、「デフォルトサービス名」として「Service#1」が登録されている。そして、「モジュール種別」として共通モジュールを示すコードである「4」が登録され、「プラットフォーム種別」として、サーバ、クライアント、ハードウェア仕様またはオペレーティングシステムを特定しないコード「0」が登録されている。

【0237】さらに、「版数」として「1」が登録され、「状態フラグ」としてリモート実行、すなわちServer#1以外のサーバによる共通モジュールServiceModule#D2の実行を示すコードである「2」が登録されている。そして、「リモート実行時のサーバアドレス」としてこの共通モジュールServiceModule#D2の起動をおこなうサーバであるServer#2のアドレスを示す「#0002」が登録され、「アクセスID」としてこのServer#2上の共通モジュールServiceModule#D2にアクセスするのに必要なID「333」が登録されている。

【0238】図26は、この場合のClient#1のクライアントデータ管理テーブルCDT1に登録された共通モジュールServiceModule#D1のサービスモジュールデータを示す説明図である。

【0239】図26において、まず、「サービスモジュール名」として「ServiceModule#D1」

が登録され、「デフォルトサービス名」として「Service#1」が登録されている。そして、「版数」として「1」が登録され、「取得サーバアドレス」としてこの共通モジュールServiceModule#D2を配信したサーバの論理アドレスを示す「#0001」が登録されている。

【0240】さらに、「状態フラグ」としてリモート実行、すなわちサーバによる共通モジュールServiceModule#D2の実行を示すコードである「2」が登録されている。そして、「リモート実行時のサーバアドレス」としてこの共通モジュールServiceModule#D2の起動をおこなうサーバであるServer#2のアドレスを示す「#0002」が登録され、「アクセスID」としてこのServer#2上の共通モジュールServiceModule#D2にアクセスするのに必要なID「333」が登録されている。

【0241】以上に説明したように、実施の形態2にかかる分散処理システムによれば、上述した「サーバにおけるモジュール実行環境移行処理」の実行によって、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールの一部を、リソースに余裕のある他のサーバ上で起動することができ、サーバのリソース不足によるスループットの低下を迅速に回避することができるとともに、効果的な負荷分散を実現することができる。

【0242】また、以上に説明した実施の形態1および2によれば、サーバからクライアントに配信されるサービスモジュールの属性（サービスモジュール名、デフォルトサービス名、版数等）や実行状態（状態フラグ）を登録設定されるサービスモジュールデータを備えたサーバデータ管理テーブルSDT1、クライアントデータ管理テーブルCDT1を、サーバ、クライアントにそれぞれ備えることで、サービスの実行に必要なサービスモジュールの実行位置が把握でき、上述した「サービスモジュールアクセス処理」を実行することによって、共通モジュールへのアクセスを確実におこなうことができる。

【0243】なお、以上に説明した原理説明、実施の形態1および2において、図6～15に示した処理を実行させるプログラム（詳しくは個別に実行可能なモジュール群）をICカードメモリ、フロッピーディスク、光磁気ディスク、CD-ROM等の記録媒体に格納し、この記録媒体に記録されたプログラムをサーバ装置およびクライアント装置にインストールすることで、上述した分散処理方法を使用し、これにより分散処理システムを構築することもできる。なお、このインストール作業は、通信回線を使用してサーバからクライアントに対して、またはサーバ間においてダウンロードすることによっておこなってもよい。

【0244】

【発明の効果】以上、説明したとおり、本発明の請求項1にかかる分散処理システムおよび請求項5にかかる分散処理方法によれば、サーバ装置は、クライアント装置の要求に応じて、クライアント装置の種別毎に異なり、かつサービスの実行に必要なモジュールをサーバ装置から取得することのできる初期設定モジュールをクライアント装置に配信し、クライアント装置は、サーバ装置からこの初期設定モジュールを取得して実行することで、クライアント装置において必要なサービスモジュールのみをサーバ装置から取得することができ、この取得したサービスモジュールを実行することにより、サービスを利用することができるので、ハードウェア仕様やオペレーティングシステムの異なるクライアント装置がネットワークに接続された場合であっても、クライアント装置に応じた初期設定モジュールを取得するだけで、サーバ装置によって提供されるサービスがクライアント装置間において差異なく利用でき、また、クライアント装置においてサービス利用手段により該当する初期設定モジュールを自動的に取得することができるので、ユーザは、クライアント装置の種別を意識することなく、サーバ装置の提供するサービスを容易に利用することができる。そして、本発明の請求項9にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項5にかかる分散処理方法を使用することができ、上記した請求項1にかかる分散処理システムの構築が可能となる。

【0245】また、本発明の請求項2にかかる分散処理システムおよび請求項6にかかる分散処理方法によれば、サーバ装置に、実行環境を問わない共通モジュールが備えられることで、サーバ装置のリソース不足状態に対して、本来、このサーバ装置において実行すべき共通モジュールを、他のサーバ装置において実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアント装置のアクセスにより、サーバ装置の資源が圧迫された場合に、サーバ装置において実行されていた共通モジュールを、リソースに余裕のある他のサーバ装置において実行できるようになり、サーバ装置のリソース不足によるスループットの低下を回避することができる。そして、本発明の請求項10にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項6にかかる分散処理方法を使用することができ、上記した請求項2にかかる分散処理システムの構築が可能となる。

【0246】また、本発明の請求項3にかかる分散処理システムおよび請求項7にかかる分散処理方法によれば、サーバ装置にサーバ装置とクライアント装置との双方の実行環境において実行可能な共通モジュールが備えられることで、クライアント装置のリソース不足状態に対して、本来、クライアント装置において実行すべき共

通モジュールを、サーバ装置において実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、他のアプリケーションプログラム等の実行により、クライアント装置の資源が圧迫された場合に、クライアント装置において実行されていた共通モジュールをサーバ装置において実行できるようになり、クライアント装置のリソース不足によるスループットの低下を回避することができる。そして、本発明の請求項11にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項7にかかる分散処理方法を使用することができ、上記した請求項3にかかる分散処理システムの構築が可能となる。

【0247】また、本発明の請求項4にかかる分散処理システムおよび請求項8にかかる分散処理方法によれば、サーバ装置が、クライアント装置に対してサービスモジュールの更新を通知することができ、さらに、その通知は、過去にそのサービスモジュールの配信をおこなった先のクライアント装置に対しておこなわれるので、クライアント装置において、ユーザが、特にサーバ装置におけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。そして、本発明の請求項12にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項8にかかる分散処理方法を使用することができ、上記した請求項4にかかる分散処理システムの構築が可能となる。

【0248】また、請求項13の発明によれば、クライアント装置に対して、クライアント装置の種別毎に異なるサービスモジュールの配信を可能とする初期設定モジュールを配信するので、クライアント装置側でサービスを利用するためのサービスプログラムに必要なサービスモジュールの変更等が生じた場合にも、初期設定モジュールの内容を変更するだけで対応でき、クライアント装置の種別毎に存在する多くのサービスモジュールの管理が容易なサーバ装置が得られるという効果を奏する。

【0249】また、請求項14の発明によれば、実行環境を問わない共通モジュールが備えられることで、リソース不足状態に対して、本来、このサーバ装置において実行すべき共通モジュールを、他のサーバ装置において実行するように実行位置を移行することができるので、リソース不足が生じた場合に、実行されていた共通モジュールを、リソースに余裕のある他のサーバ装置において実行できるようになり、これによりスループットの低下を回避することが可能なサーバ装置が得られるという効果を奏する。

【0250】また、請求項15の発明によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、他のサーバ装置に対する起動要求の対象とするので、起動要求先のサーバ

装置のリソースへの大きな圧迫を回避することが可能なサーバ装置が得られるという効果を奏する。

【0251】また、請求項16の発明によれば、起動要求された共通モジュールへのアクセスを、起動要求を発したサーバ装置だけでなく、その共通モジュールの実行を必要とするクライアント装置に対しても直接に可能とするので、起動要求された共通モジュールの利用を容易にするサーバ装置が得られるという効果を奏する。

【0252】また、請求項17の発明によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスを容易にするサーバ装置が得られるという効果を奏する。

【0253】また、請求項18の発明によれば、モジュール配信先テーブルを参照することによって、更新されたサービスモジュールの取得が必要なクライアント装置のみに更新通知をおこなうので、効率的なサービスモジュールの更新処理が可能なサーバ装置が得られるという効果を奏する。

【0254】また、請求項19の発明によれば、サービス利用手段によって、サーバ装置から初期設定モジュールを取得して実行するだけで、この初期設定モジュールの処理により、サービスを利用するのに必要なサービスモジュールのみをサーバ装置から取得することができるので、サーバ装置によって提供されるサービスが差異なく利用できるとともに、ユーザが、クライアント装置の種別を意識することなく、サーバ装置の提供するサービスを容易に利用することができるクライアント装置が得られるという効果を奏する。

【0255】また、請求項20の発明によれば、サービス利用手段において、クライアント装置の機能と処理能力を調査する調査機構が設けられ、初期設定モジュールの実行時にこの調査機構を実行して取得される調査結果に応じて、サーバ装置から取得するサービスモジュールを選択することができるので、調査結果が処理能力の低下を示す場合に、実行可能なサービスモジュールだけを取得して実行し、サービス実行に必要な他のサービスモジュールの実行をサーバ装置等に依頼することが可能になり、これによりストレスのないサービス利用が可能なクライアント装置が得られるという効果を奏する。

【0256】また、請求項21によれば、クライアントデータ管理テーブルに、サービスを利用するのに必要なサービスモジュールが登録されていない場合に、そのサービスモジュールをサーバ装置から取得して、クライアントデータ管理テーブルに登録するので、サービスの利用に必要な最低限のサービスモジュールだけを付属する記憶装置等に備えておくことができ、資源の有効な活用を実現することができるクライアント装置が得られるという効果を奏する。

【0257】また、請求項22によれば、実行環境を問

わない共通モジュールを実行している状態におけるリソース不足状態に対して、本来、このクライアント装置において実行すべき共通モジュールを、サーバ装置において実行するように実行位置を移行することができるのでリソース不足が生じた場合に、実行されていた共通モジュールを、リソースに余裕のあるサーバ装置において実行できるようになり、これによりスループットの低下を回避することができるクライアント装置が得られるという効果を奏する。

【0258】また、請求項23によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、サーバ装置に対する起動要求の対象とするので、起動要求先のサーバ装置のリソースへの大きな圧迫を回避することができるクライアント装置が得られるという効果を奏する。

【0259】また、請求項24によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスが容易なクライアント装置が得られるという効果を奏する。

【0260】また、請求項25によれば、更新モジュール取得手段によって、サーバ装置からサービスモジュールの更新が通知された際に、更新されたサービスモジュールを取得するので、ユーザ自らが、サーバ装置におけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能なクライアント装置が得られるという効果を奏する。

【図面の簡単な説明】

【図1】本発明にかかる分散処理システムにおいて、提供されるサービス毎に用意されたサービスデータのモジュール構成を示す説明図である。

【図2】本発明にかかるサーバのシステム実行形態を示す説明図である。

【図3】本発明にかかるサーバデータ管理テーブルの各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【図4】本発明にかかるクライアントのシステム実行形態を示す説明図である。

【図5】本発明にかかるクライアントデータ管理テーブルの各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【図6】本発明にかかるクライアントデーモンの起動において実行される処理を示したフローチャートである。

【図7】本発明にかかるサーバデーモンの起動時に実行される処理を示したフローチャートである。

【図8】本発明にかかるシステム監視スレッドによる処理を示したフローチャートである。

【図9】本発明にかかる初期設定モジュール取得処理を示したフローチャートである。

【図10】本発明にかかる初期設定モジュールによる処

理を示したフローチャートである。

【図11】本発明にかかるサービス起動処理を示したフローチャートである。

【図12】本発明にかかるクライアントにおけるモジュール実行環境移行処理を示したフローチャートである。

【図13】本発明にかかるサービスモジュールアクセス処理を示したフローチャートである。

【図14】本発明にかかるサーバにおけるモジュール実行環境移行処理を示したフローチャートである。

【図15】本発明にかかるサービスモジュール更新処理 10を示したフローチャートである。

【図16】実施の形態1にかかる分散処理システムのサービス未起動時のシステム構成を示した説明図である。

【図17】実施の形態1において、Service#1を実行するために必要なサービスデータを示す説明図である。

【図18】実施の形態1にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。

【図19】実施の形態1において、Server#1の初期設定モジュールServiceModule#C1 20のサービスモジュールデータを示す説明図である。

【図20】実施の形態1において、Client#1の共通モジュールServiceModule#D1のサ*

*サービスモジュールデータを示す説明図である。

【図21】実施の形態1にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。

【図22】実施の形態1において、Server#1の共通モジュールServiceModule#D1のサービスモジュールデータを示す説明図である。

【図23】実施の形態1において、Client#1の共通モジュールServiceModule#D1のサービスモジュールデータを示す説明図である。

【図24】実施の形態2にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。

【図25】実施の形態2において、Server#1の共通モジュールServiceModule#D1のサービスモジュールデータを示す説明図である。

【図26】実施の形態2において、Client#1の共通モジュールServiceModule#D1のサービスモジュールデータを示す説明図である。

【符号の説明】

SD1, SD2 サーバデーモン

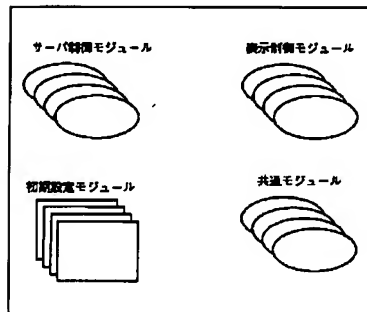
CD1 クライアントデーモン

SDT1, SDT2 サーバデータ管理テーブル

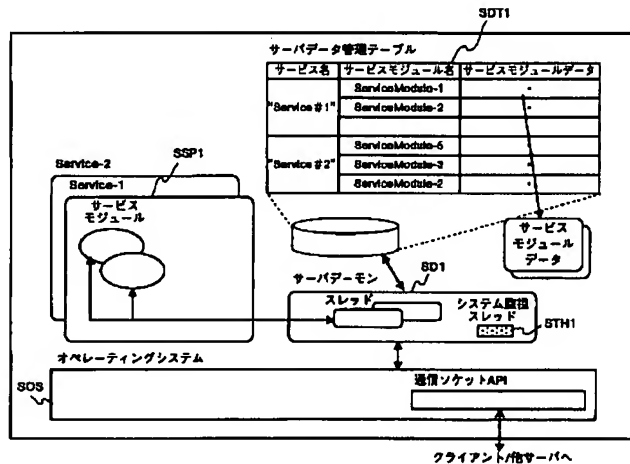
CDT1 クライアントデータ管理テーブル

STH1, STH2, CTH1 システム監視スレッド

【図1】



【図2】



【図5】

サービスモジュール名:

デフォルトサービス名:

版数: 取得サーバアドレス:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図20】

サービスモジュール名:

デフォルトサービス名:

版数: 取得サーバアドレス:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図3】

サービスモジュール名:

デフォルトサービス名:

配送先クライアントアドレスリスト:

モジュール識別:

プラットフォーム識別:

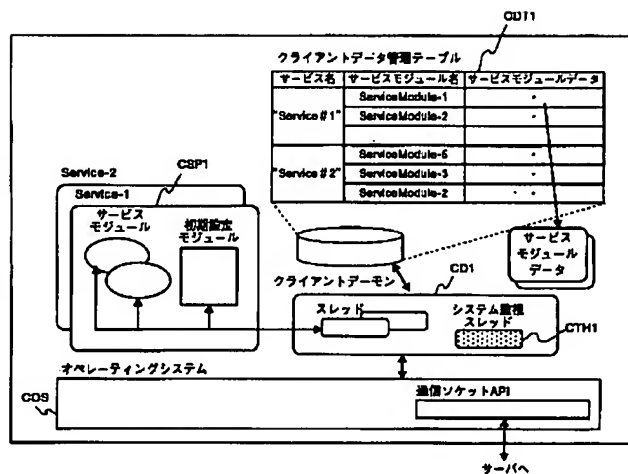
版数:

状態フラグ:

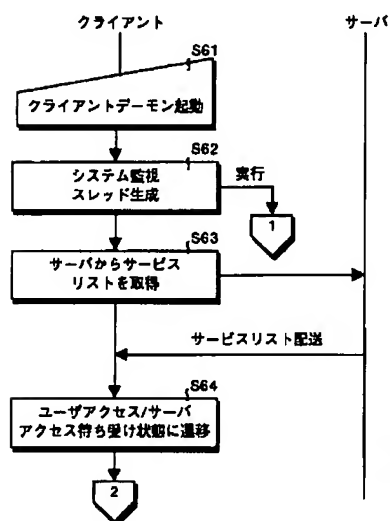
ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

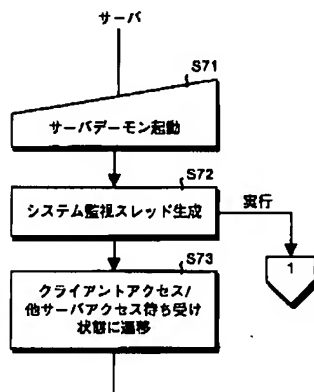
【図4】



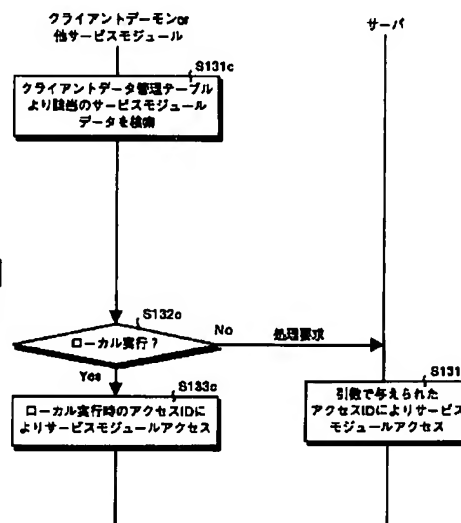
【図6】



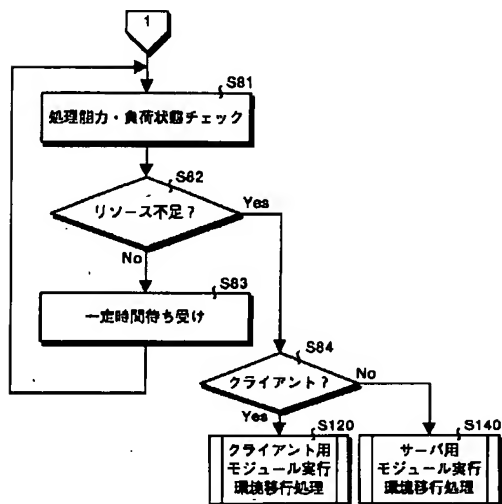
【図7】



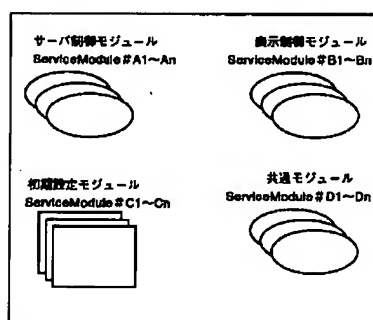
【図13】



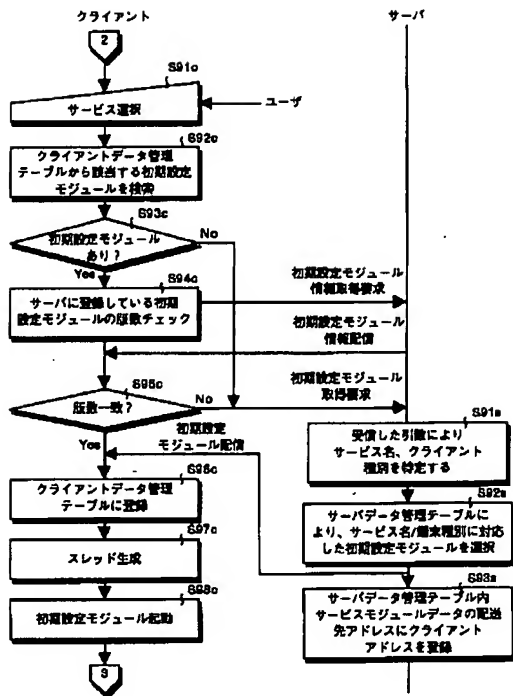
【図8】



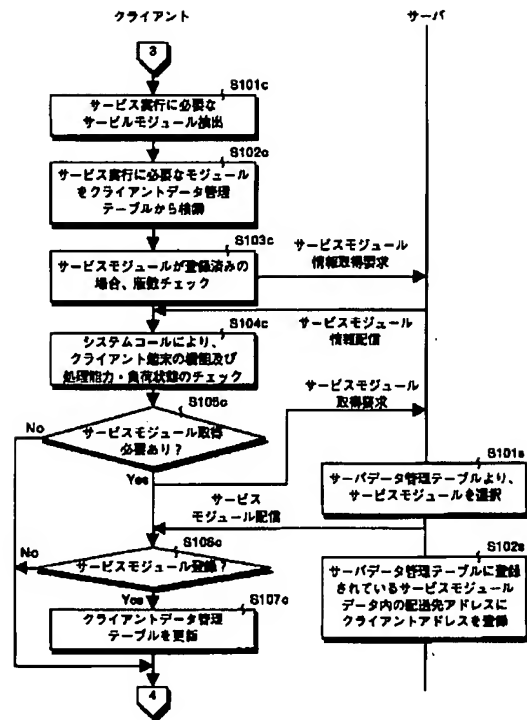
【図17】



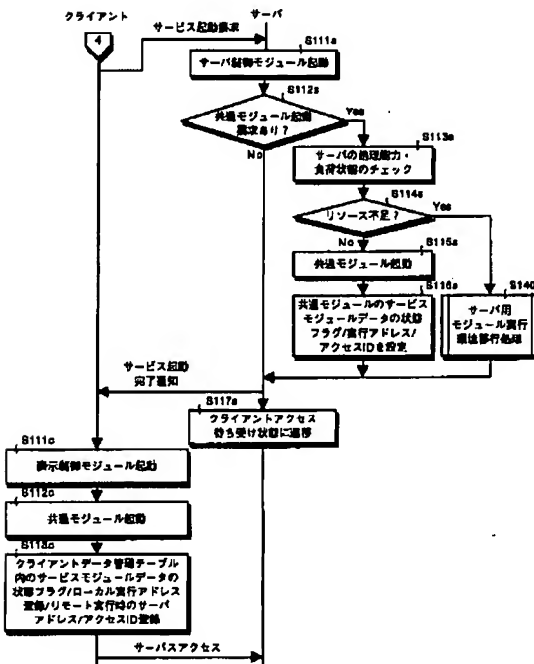
【図9】



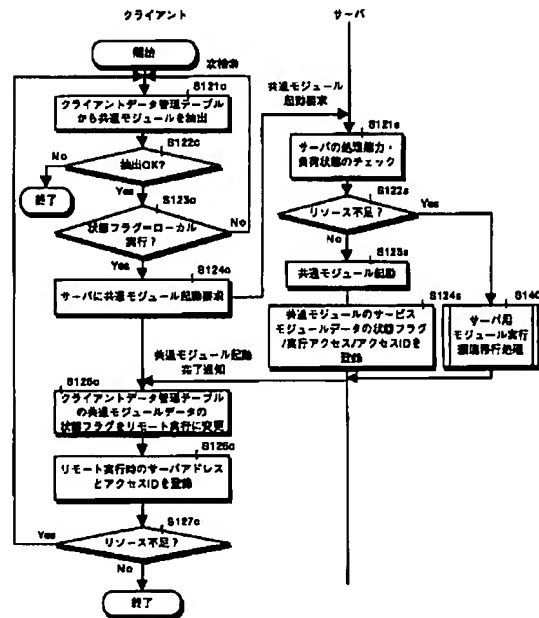
【図10】



【図11】



【図12】



【図19】

サービスモジュール名:

デフォルトサービス名:

配送先クライアントアドレスリスト:

モジュール種別:

プラットフォーム種別:

版数:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図22】

サービスモジュール名:

デフォルトサービス名:

配送先クライアントアドレスリスト:

モジュール種別:

プラットフォーム種別:

版数:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図23】

サービスモジュール名:

デフォルトサービス名:

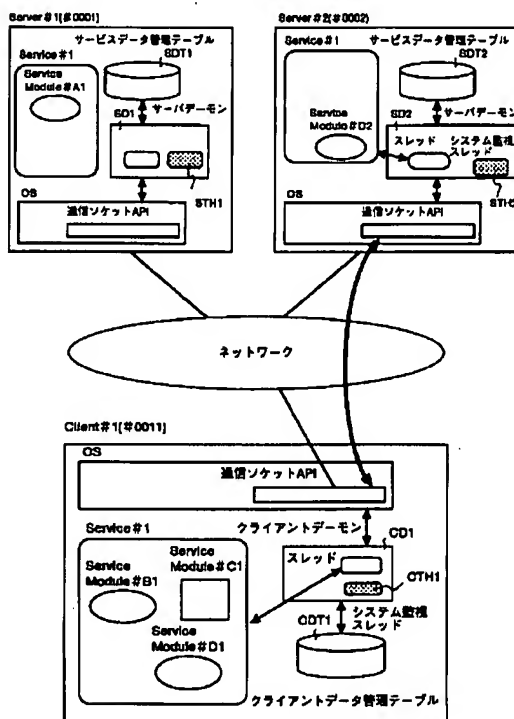
版数: 取得サーバアドレス:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図24】



【図25】

サービスモジュール名:

デフォルトサービス名:

配送先クライアントアドレスリスト:

モジュール種別:

プラットフォーム種別:

版数:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図26】

サービスモジュール名:

デフォルトサービス名:

版数: 取得サーバアドレス:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【手続補正書】

【提出日】平成11年8月20日（1999. 8. 20）

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】請求項4

【補正方法】変更

【補正内容】

【請求項4】 前記サーバ装置は、前記モジュールが更新された際に、該モジュールを配信した先のクライアント装置に、前記更新されたモジュールの更新を通知する更新通知手段を備え、
前記クライアント装置は、前記更新されたモジュールの更新が通知された場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得手段を備えたことを特徴とする請求項1～3のいずれか一つに記載の分散処理システム。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】請求項10

【補正方法】変更

【補正内容】

【請求項10】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、
前記サーバ装置と他のサーバ装置との双方の実行環境において実行可能な共通モジュールを前記サーバ装置において実行するサーバ共通モジュール実行ステップと、
前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップと、
を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0044

【補正方法】変更

【補正内容】

【0044】また、請求項10の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバと他のサーバとの双方の実行環境において実行可能な共通モジュールをサーバにおいて実行するサーバ共通モジュール実行ステップ（「サービス起動処理」のステップS115s、
「クライアントにおけるモジュール実行環境移行処理」

のステップS123s、または「サーバにおけるモジュール実行環境移行処理」のステップS143sに相当）と、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、他のサーバに対してその他のサーバ側での共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップ（「サーバにおけるモジュール実行環境移行処理」のステップS145s1に相当）と、を実行させることを特徴とする。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0206

【補正方法】変更

【補正内容】

【0206】図18は、実施の形態1にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。図18において、Server#1は、図17に示す状態に加えて、Service#1を提供するサーバ用サービスプログラムとして、サーバ制御モジュールServiceModule#A1、共通モジュールServiceModule#D2が用意され、これらサービスモジュールの実行においてサーバデーモンSD1内にスレッドが生成される。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0207

【補正方法】変更

【補正内容】

【0207】また、Client#1は、図17に示す状態に加えて、Service#1を提供するクライアント用サービスプログラムとして、表示制御モジュールServiceModule#B1、初期設定モジュールServiceModule#C1、共通モジュールServiceModule#D1が用意され、これらサービスモジュールの実行においてクライアントデーモンCD1内にスレッドが生成される。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0218

【補正方法】変更

【補正内容】

【0218】図22において、まず、「サービスモジュール名」として「ServiceModule#D1」が登録され、「デフォルトサービス名」として「Service#1」が登録されている。そして、「配送先クライアントリスト」には、上記したClient#1の論理アドレスである「#0011」が登録されている。

【手続補正7】

【補正対象書類名】明細書

【補正対象項目名】0231

【補正方法】変更

【補正内容】

【0231】ここでは、図14に示したモジュール実行環境移行処理における第2のサーバが、Server #2に該当し、このServer #2によって、Client #1から起動要求のあった共通モジュールServiceModule #D2の起動がおこなわれる。よって、図24に示すように、Server #2においては、Service #1を提供するサーバ用サービスプログラムとして、共通モジュールServiceModule #D2が用意され、このサービスモジュールの実行においてサーバデーモンSD2内にスレッドが生成された状態にある。

【手続補正8】

【補正対象書類名】明細書

【補正対象項目名】0238

【補正方法】変更

【補正内容】

【0238】図26は、この場合のClient #1のクライアントデータ管理テーブルCDT1に登録された共通モジュールServiceModule #D2のサービスモジュールデータを示す説明図である。

【手続補正9】

【補正対象書類名】明細書

【補正対象項目名】0239

【補正方法】変更

【補正内容】

【0239】図26において、まず、「サービスモジュール名」として「ServiceModule #D2」が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「版数」として「1」が登録され、「取得サーバアドレス」としてこの共通モジュールServiceModule #D2を配信したサーバの論理アドレスを示す「#0001」が登録されている。

【手続補正10】

【補正対象書類名】明細書

【補正対象項目名】図25

【補正方法】変更

* 【補正内容】

【図25】実施の形態2において、Server #1の共通モジュールServiceModule #D2のサービスモジュールデータを示す説明図である。

【手続補正11】

【補正対象書類名】明細書

【補正対象項目名】図26

【補正方法】変更

【補正内容】

【図26】実施の形態2において、Client #1の共通モジュールServiceModule #D2のサービスモジュールデータを示す説明図である。

【手続補正12】

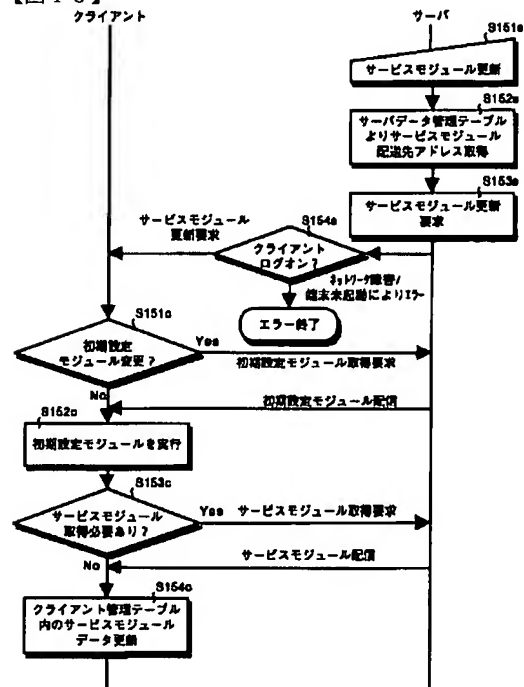
【補正対象書類名】図面

【補正対象項目名】図15

【補正方法】変更

【補正内容】

【図15】



*

フロントページの続き

Fターム(参考) 5B045 GG01 GG02

5B089 GA11 GA21 JA11 JB07 JB15

JB18 KA06 KC11 KC15